Explainability in Graph Neural Networks

Jimyeung Seo

Graph & Language Intelligence Laboratory
Department of Computer Science and Engineering
Konkuk University

2026.01.06

CONTENTS

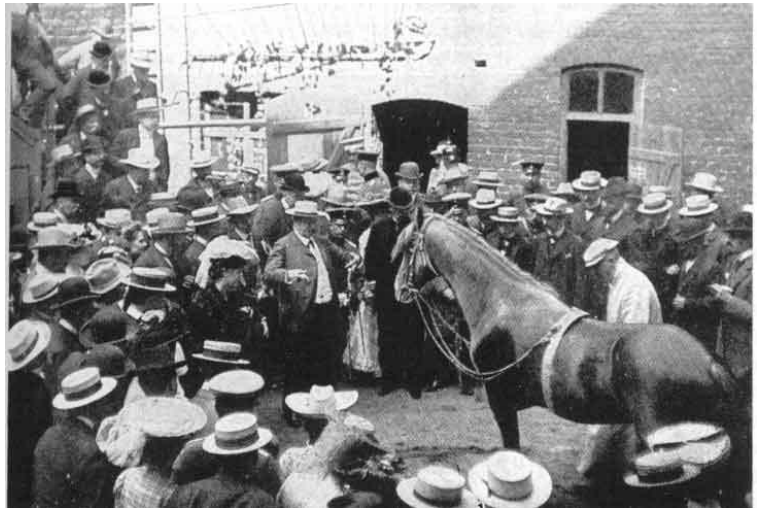
1. What is Explainability?
2. Explainable GNNs
3. Identifying Informative Substructures
4. Conclusion

CONTENTS

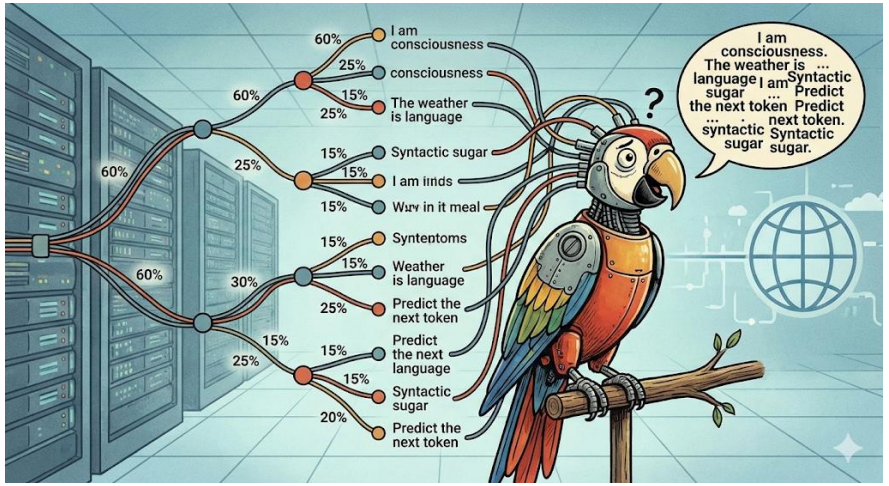
1. What is Explainability?
   - The “Black-box” problem
   - Challenges with graphs
2. Explainable GNNs
3. Identifying Informative Substructures
4. Conclusion

Today in ML

- Clever Hans Effect
  - ✓ 본질(Causal feature)이 아닌 spurious feature, noise를 보고 예측



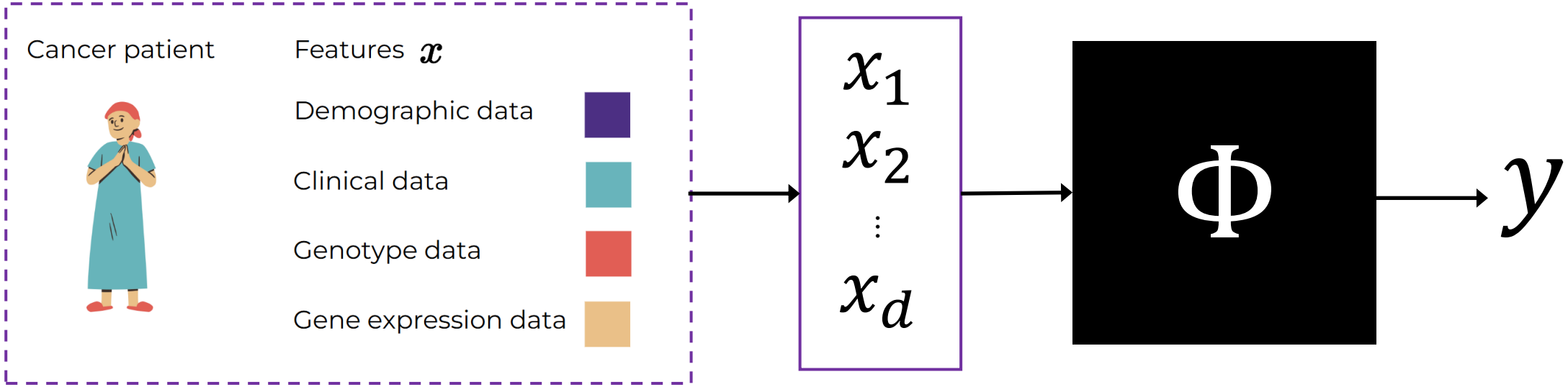
- Stochastic Parrots [1]
  - ✓ LLM은 의미를 이해하고 말하는 것이 아니라, 그저 방대한 데이터에서 통계적으로 다음에 올 법한 단어를 조합해서 내뱉을 뿐



[1] Bender and Gebru, et al, “On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?”, FAccT’21

Today in ML

- Lack of Transparency (The Black-box problem)



Why do we need?

- Trust
  - ✓ 의료, 금융, 법률과 같은 High-Stakes Domains에서는 모델의 예측 근거가 인간의 상식이나 도메인 지식과 일치하는지 검증되어야 함

- Model Debugging & Improvement
  - ✓ **가짜 상관관계(Spurious correlation)**을 탐지함으로써 데이터의 편향을 학습하고 있음을 발견하면, 이를 역으로 이용해 데이터를 정제하거나 모델 구조를 개선할 수 있음

- Scientific Discovery
  - ✓ 신약 개발, 단백질 구조 분석 등에서 설명 가능성은 단순히 모델 해석을 넘어 Knowledge Discovery의 수단

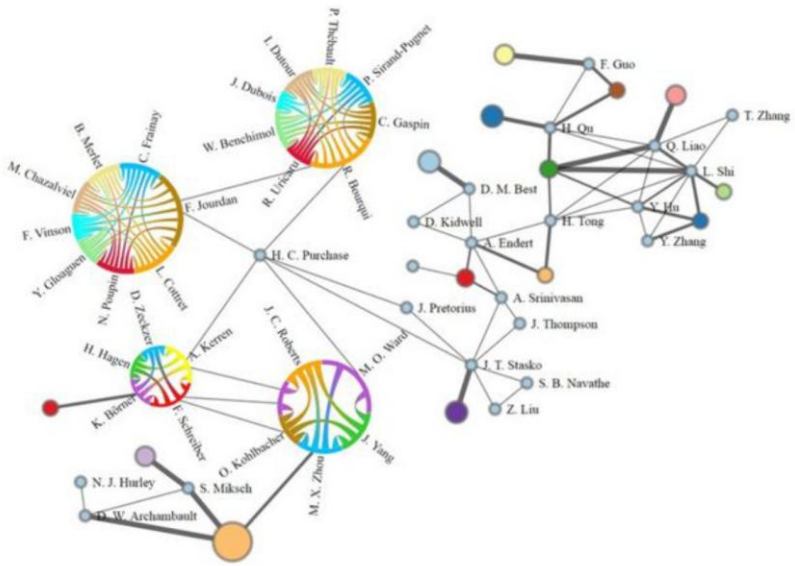
- Robustness & Generalization
  - ✓ 쓸데없는 허위 정보는 무시하고, **중요한 본질을 학습한 모델은 out-of-distribution에도 성능이 안정적**
  - ✓ Causal factor에 집중하도록 학습된 모델은 데이터 분포가 바뀌어도 성능이 급격히 떨어지지 않음

Explainablity vs. Interpretability

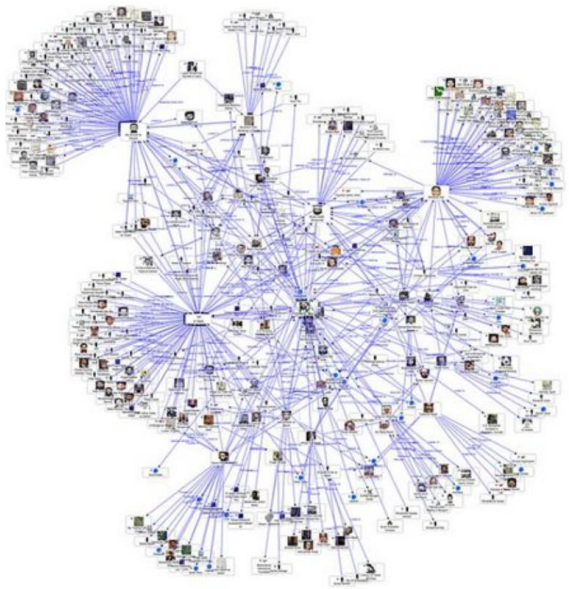
- Interpretable Models: 다른 보조 도구/기술 없이도 사람이 이해할 수 있음
- Explainable Models: 추가적인 기술이 있어야 사람이 이해할 수 있음

Challenges with graphs

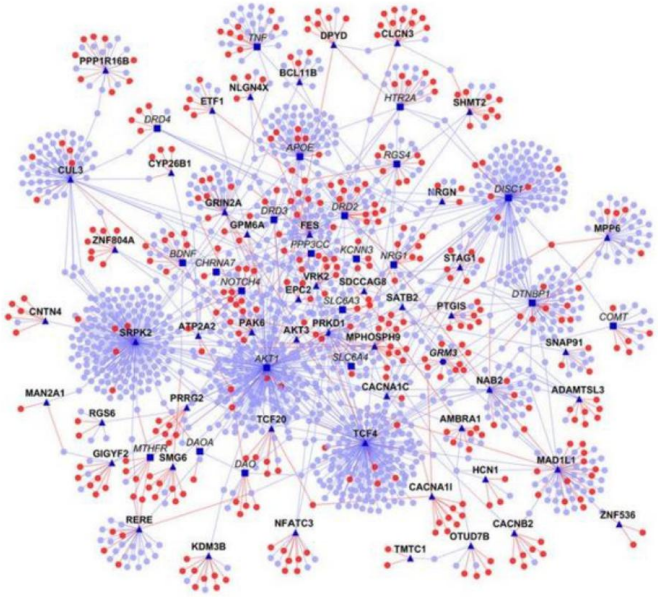
- Variable topology: 다양한 구조에 적용 가능하면서 충분히 표현력 높은 신경망을 설계하기 어려움
- Huge graphs: 수 백만개의 노드와 수 십조개의 엣지가 존재



Co-authorship Networks



Social Networks



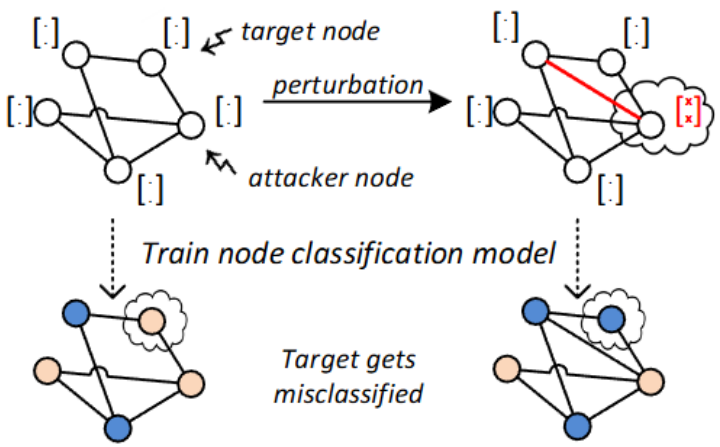
Protein interactions

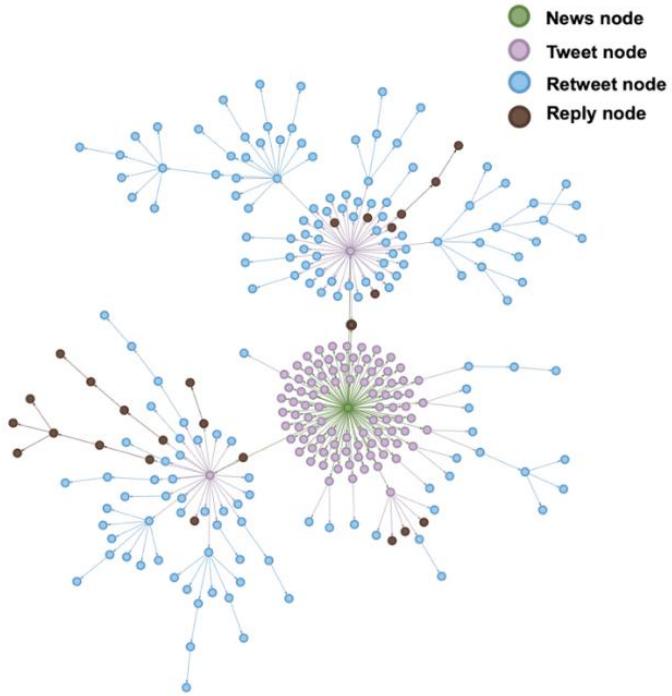
Challenges with graphs

- CNN은 픽셀 몇 개가 바뀌어도 의미가 유지되는 경우가 많지만, GNN은 edge perturbation이 곧 의미/성능/속성의 급격한 변화로 이어질 수 있음

- 중요 노드/엣지는 서로 연결되어 있지 않을 수 있어 구조적으로 연관된 substructure 단위의 설명이 필요

[1]

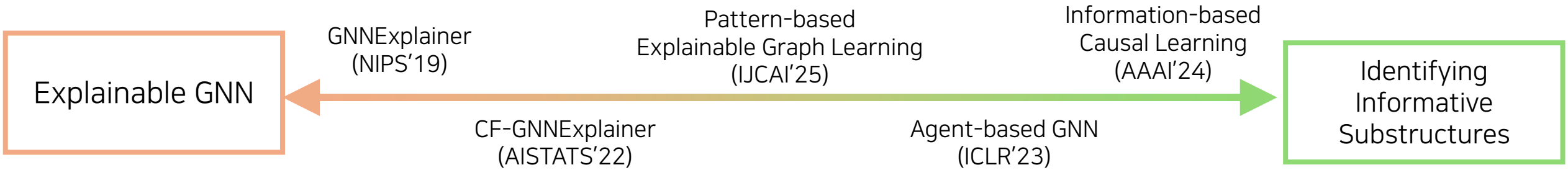




[1] Adversarial Attacks on Neural Networks for Graph Data, IJCAI'19

Overview

- Explainable GNN(XAI)
- Identifying Informative Substructures

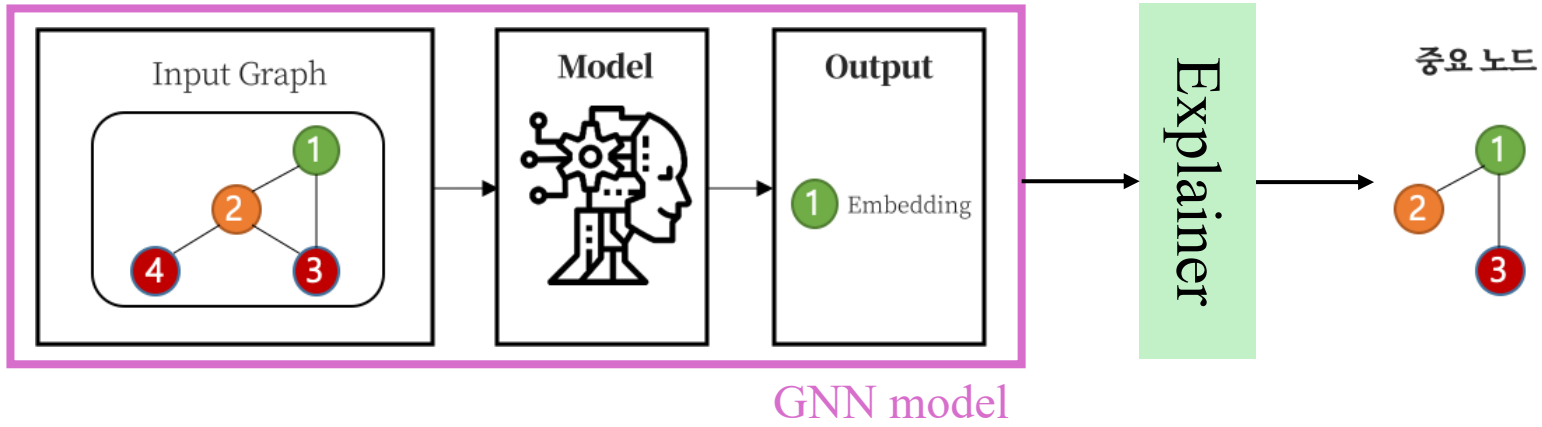


CONTENTS

1. What is Explainability?
2. Explainable GNNs
   - Factual explanations
   - Counterfactual explanation
3. Identifying Informative Substructures
4. Conclusion

Explainable GNNs

- 목적: 특정 노드를 classification함에 있어 영향을 미친 노드들, 즉 subgraph를 찾는 것



- Factual explanations
  - ✓ 전체 그래프와 subgraph는 동일한 label을 가짐
  - ✓ Subgraph는 전체 그래프보다 작아야 함

- Counterfactual explanations
  - ✓ 해당 subgraph를 제거하고 남은 부분은 기존과 정반대 클래스를 가져야 함

Factual Explanations: GNNExplainer (NIPS'19)

Motivation

- 기존 ML/DL에서 사용했던 방식은 그래프에 직접 적용할 수 없음
  - ✓ 모델을 더 단순한 proxy model로 근사화한 후 이를 통해 설명을 찾음
  - ✓ 모델의 high level features에 대한 해석을 제시
  - ✓ 영향력 있는 input instances를 식별

- 이러한 접근법들은 그래프의 본질인 관계 정보를 통합하지 못함
  - ✓ 그래프 기반 ML은 이 부분이 결정적이기 때문에 GNN 예측에 대한 설명은 반드시 그래프가 제공하는 풍부한 관계 정보와 노드 특징을 모두 활용해야 함

Factual Explanations: GNNExplainer (NIPS'19)

GNNExplainer: Generating Explanations for Graph Neural Networks

Rex Ying[†] Dylan Bourgeois[†,‡] Jiaxuan You[†] Marinka Zitnik[†] Jure Leskovec[†]

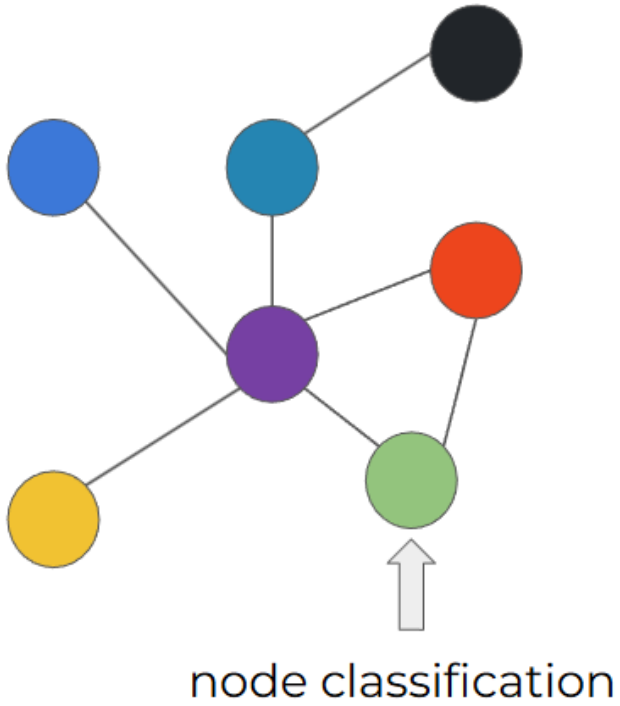
[†]Department of Computer Science, Stanford University
[‡]Robust.AI
{rexying, dtsbourg, jiaxuan, marinka, jure}@cs.stanford.edu

Cited: 2610

- Challenges
  - ✓ C1: 하나의 노드 임베딩을 구성할 때, 어떤 이웃 노드들이 영향을 미쳤는지
  - ✓ C2: 노드 내의 어떤 features가 영향을 미쳤는지

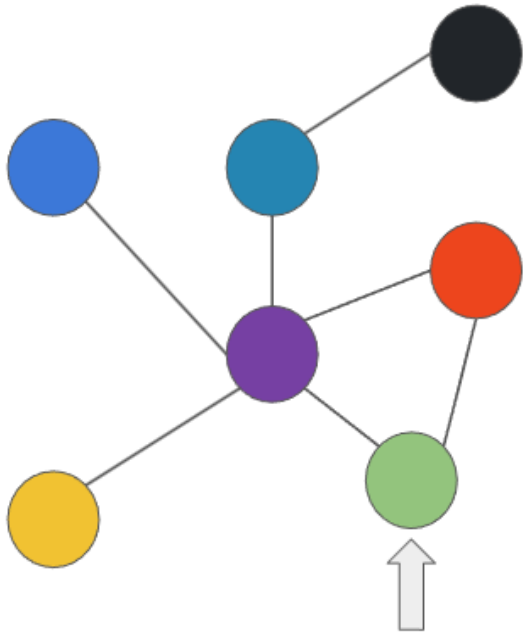
Factual Explanations: GNNExplainer (NIPS'19)

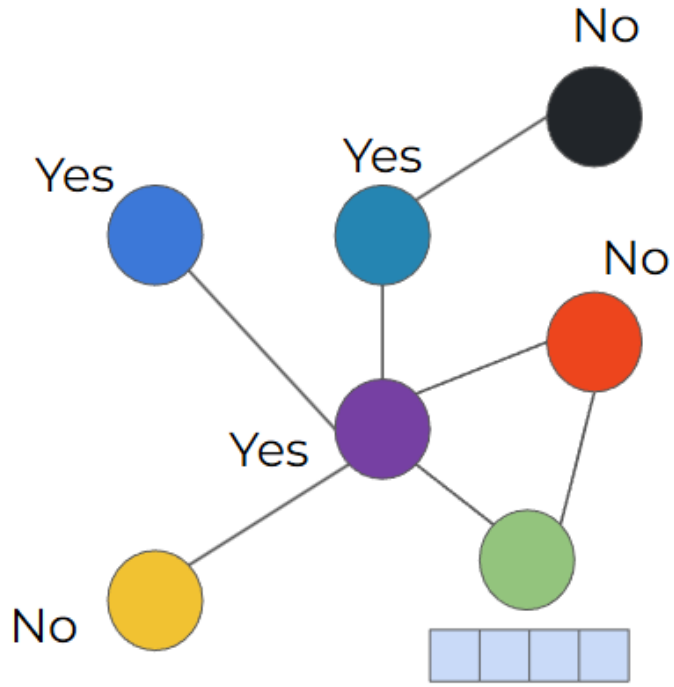
- C1: 하나의 노드 임베딩을 구성할 때, 어떤 이웃 노드들이 영향을 미쳤는지



Factual Explanations: GNNExplainer (NIPS'19)

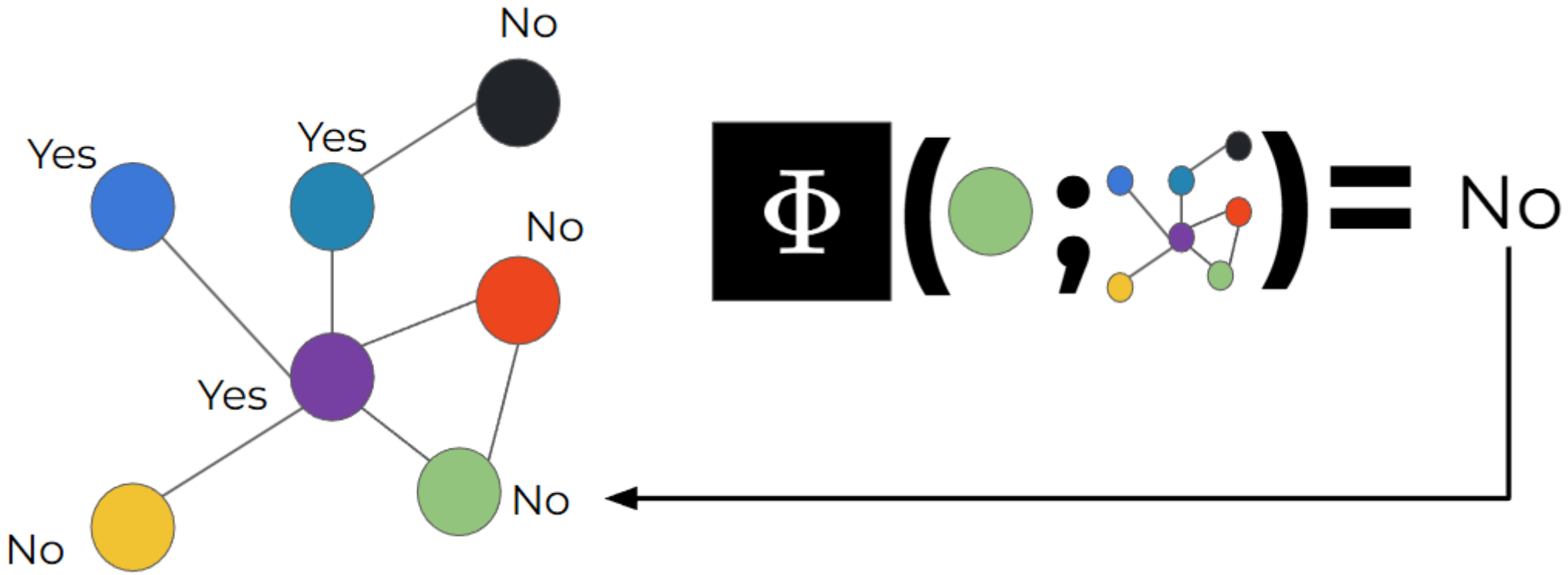
- C1: 하나의 노드 임베딩을 구성할 때, 어떤 이웃 노드들이 영향을 미쳤는지





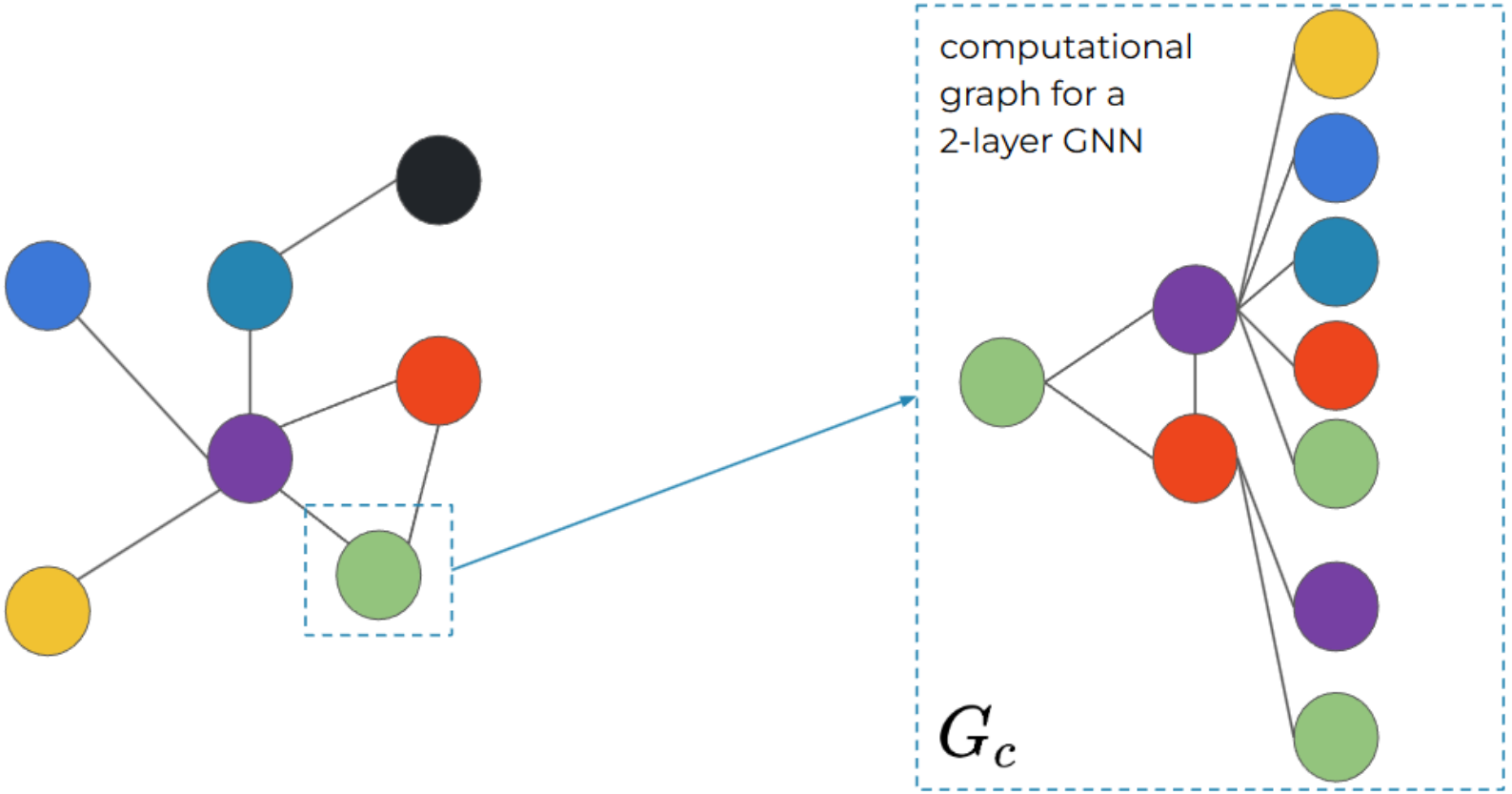
Factual Explanations: GNNExplainer (NIPS'19)

- C1: 하나의 노드 임베딩을 구성할 때, 어떤 이웃 노드들이 영향을 미쳤는지



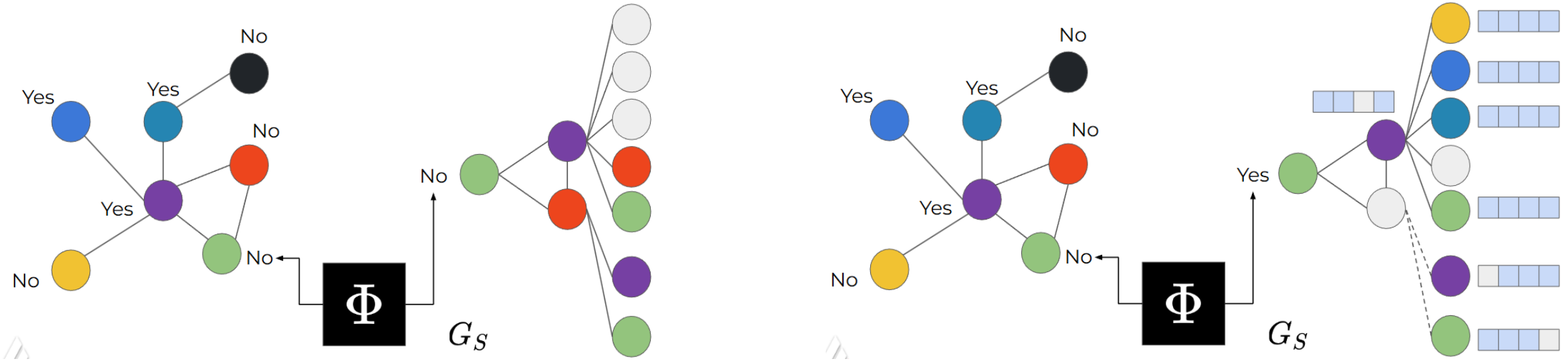
Factual Explanations: GNNExplainer (NIPS'19)

- C1: 하나의 노드 임베딩을 구성할 때, 어떤 이웃 노드들이 영향을 미쳤는지



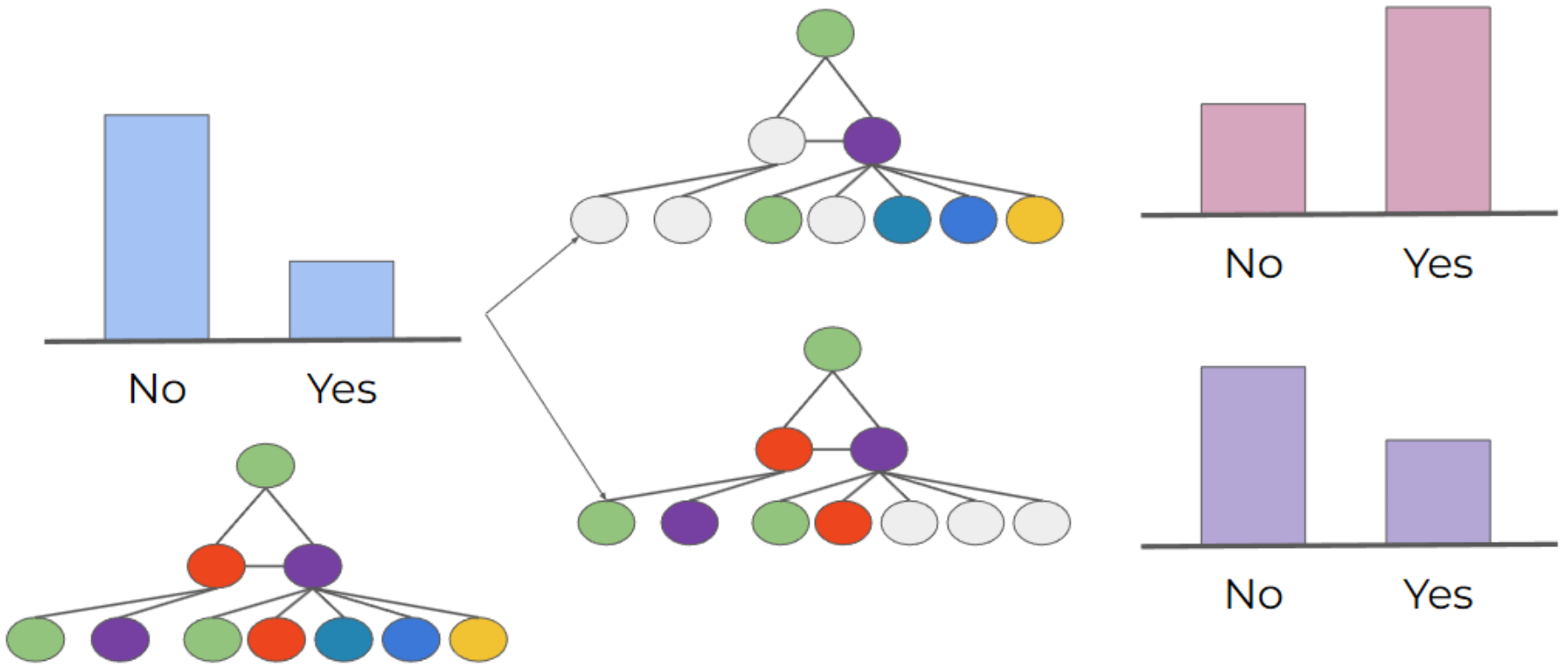
Factual Explanations: GNNExplainer (NIPS'19)

- C1: 하나의 노드 임베딩을 구성할 때, 어떤 이웃 노드들이 영향을 미쳤는지



Factual Explanations: GNNExplainer (NIPS'19)

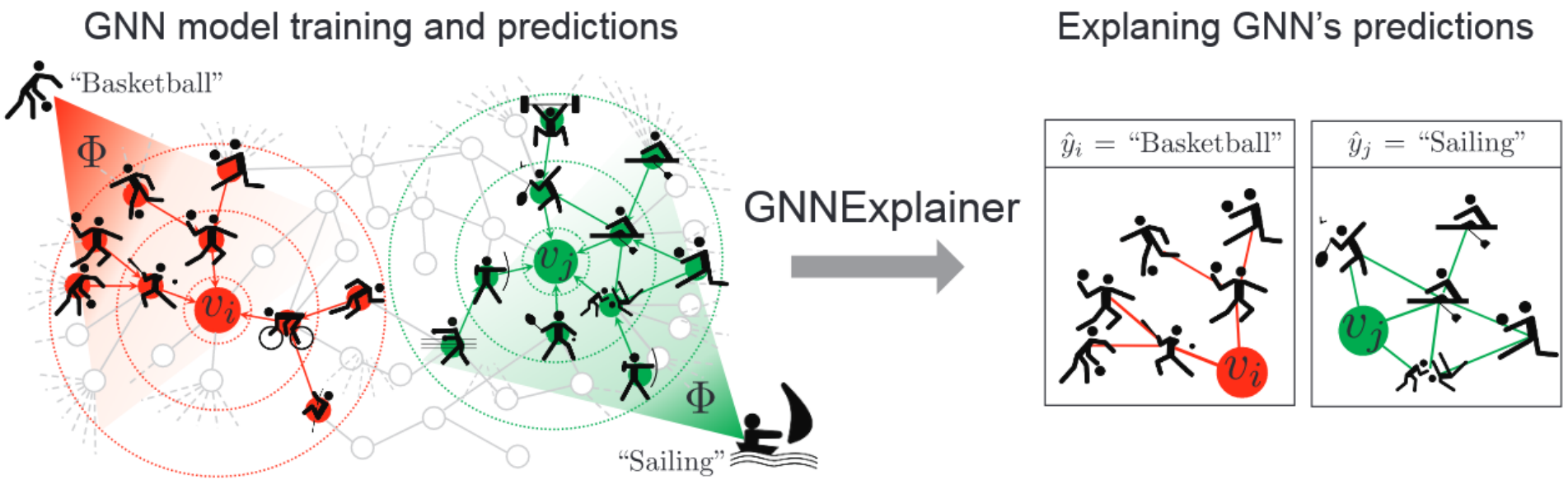
- C1: 하나의 노드 임베딩을 구성할 때, 어떤 이웃 노드들이 영향을 미쳤는지



Factual Explanations: GNNExplainer (NIPS'19)

- C1: 하나의 노드 임베딩을 구성할 때, 어떤 이웃 노드들이 영향을 미쳤는지
  - ✓ Full graph만큼의 정보를 가지고 있는 subgraph를 탐색
  - ✓ Mutual Information이 최대인 subgraph를 구함

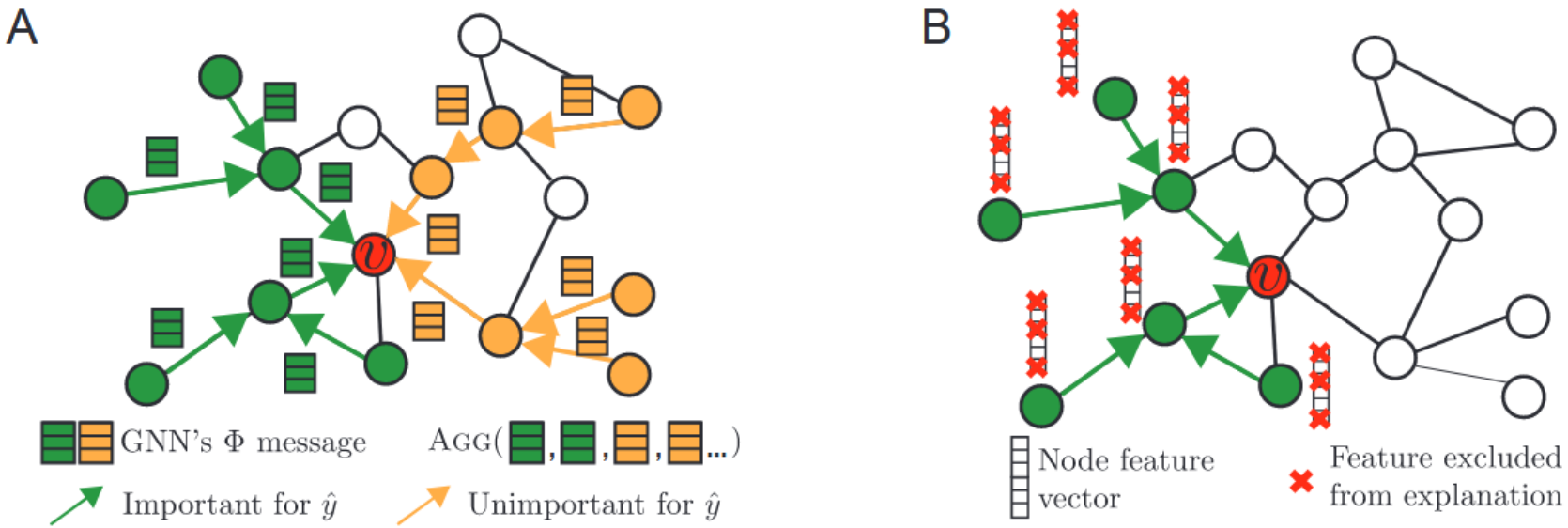
$$\max_{G_S} MI\left(Y, (G_S, X_S)\right) = H(Y) - H(Y|G = G_S, X = X_S).$$



Factual Explanations: GNNExplainer (NIPS'19)

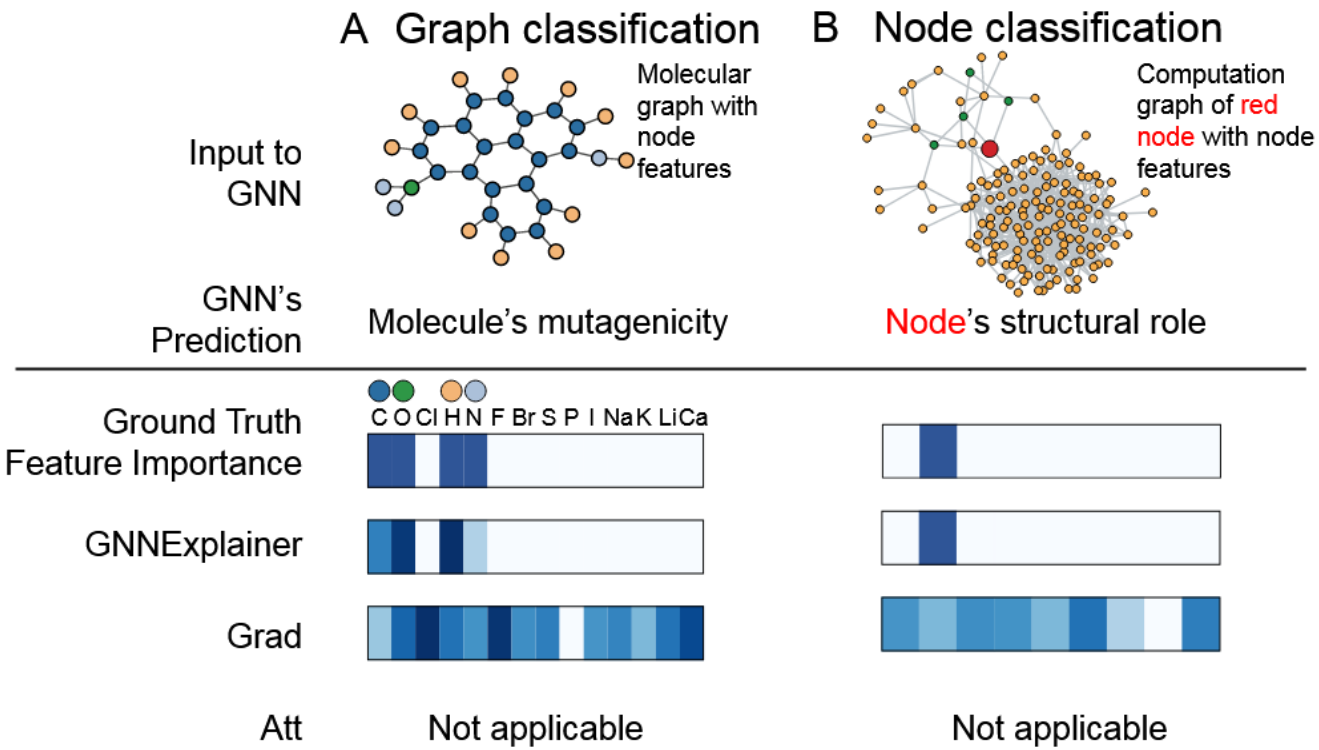
- C2: 노드 내의 어떤 features가 영향을 미쳤는지
  - ✓ Joint Learning: Feature Selector F를 학습
  - ✓ 이전에는 G_S에 포함된 모든 features를 사용했다면 이제는 여기에 mask를 추가한 X_S^F를 생성

$$\max_{G_S,F} MI\left(Y,(G_S,F)\right) = H(Y) - H(Y|G=G_S, X=X_S^F),$$



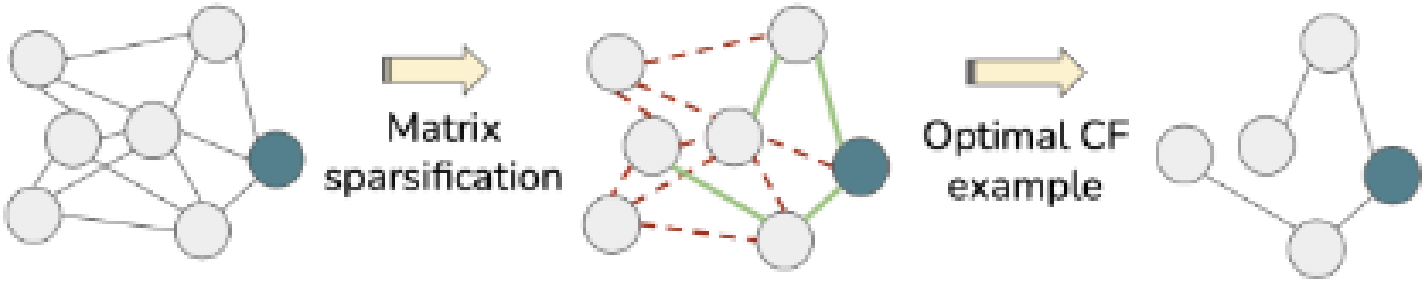
Factual Explanations: GNNExplainer (NIPS'19)

- Visualization from MUTAG and BA-COMMUNITY



Counterfactual Explanations

- Motivations
  - ✓ Factual explainers는 subgraph의 크기를 사용자가 지정해야 해서 자동으로 minimial subgraph를 식별할 수 없음
  - ✓ Counterfactual explanation을 활용하면 GNN 모델의 예측을 뒤바꿀 수 있는 최소 변경 사항을 식별할 수 있음



Counterfactual Explanations: CF-GNNExplainer (AISTATS'22)

CF-GNNExplainer: Counterfactual Explanations for Graph Neural Networks

Ana Lucic [1] **Maartje ter Hoeve** [1] **Gabriele Tolomei** [2] **Maarten de Rijke** [1] **Fabrizio Silvestri** [2]
[1] University of Amsterdam [2] Sapienza University of Rome

Cited: 268

- Goal: 예측 결과를 뒤집기 위한 최소한의 변화를 찾는 것
  - ✓ Prediction Loss: 예측 결과가 원래와 달라야 함
  - ✓ Distance Loss: 원본 그래프와 최대한 비슷해야 함

$$\mathcal{L} = \mathcal{L}_{pred}(v, \bar{v} \mid f, g) + \beta \mathcal{L}_{dist}(v, \bar{v} \mid d),$$

Explainable GNNs

- Limitations
  - ✓ 그래프 하나를 설명할 때마다 새로 학습해야 하며, 데이터셋 전반적인 설명을 하지 못함 (Instance-level)

- Recent works
  - ✓ MAGE (ICLR'25) [1]
    - 자주 등장하는 부분 구조를 조립하여 설명을 생성
    - VAE 기반의 생성 모델을 사용하여, 특정 클래스가 될 확률을 최대화하는 "가장 이상적인 motif"를 만들어냄
  - ✓ G-Refer (WWW'25) [2]
    - 설명가능한 추천 연구에서 GNN은 명시적이고 의미론적으로 풍부한 협업 정보를 포착하는 데 한계가 있었음
    - Hybrid graph retrieval을 통해 user-item interaction graphs에서 구조적/의미적 CF signal을 검색
    - 검색된 CF 정보는 LLM을 통해 human-understandable text로 변환됨
  - ✓ CoED GNN (ICLR'25) [3]
    - 기존 GNN의 undirected 가정 때문에 oversmoothing 문제가 있고 edge를 0/1로만 다뤄서 최적화가 어려움
    - Edge에 고정된 방향이 아닌 연속적인 방향성을 학습 variable로 설정하여 정보가 흘러가는 Optimal flow를 자연스럽게 학습함

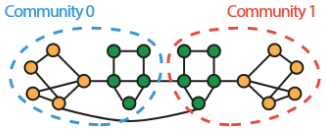
[1] MAGE: Model-Level Graph Neural Networks Explanations via Motif-based Graph Generation, ICLR'25
[2] G-Refer: Graph Retrieval-Augmented Large Language Model for Explainable Recommendation, WWW'25
[3] Improving Graph Neural Networks by Learning Continuous Edge Directions, ICLR'25

Is it done?

- Explainable GNN methods cannot change overall task performance
  - ✓ Post-hoc 해석 도구일 뿐, GNN 아키텍처를 변환시키지 않음

Is it done?

- Explainable GNN methods cannot change overall task performance
  - ✓ Post-hoc 해석 도구일 뿐, GNN 아키텍처를 변환시키지 않음

	BA-Shapes	BA-Community	Tree-Cycles	Tree-Grid
Base				
Motif				
Node Features	None	$\mathcal{N}(\mu_l, \sigma_l)$ where l = community ID	None	None
Explanation content	Graph structure	Graph structure Node feature information	Graph structure	Graph structure
Explanation accuracy				
Att	0.815	0.739	0.824	0.612
Grad	0.882	0.750	0.905	0.667
GNNExplainer	**0.925**	**0.836**	**0.948**	**0.875**

Method	REE-CYCLES *Fid.* ▼	*Size* ▼	*Spars.* ▲	*Acc.* ▲	TREE-GRID *Fid.* ▼	*Size* ▼	*Spars.* ▲	*Acc.* ▲	BA-SHAPES *Fid.* ▼	*Size* ▼	*Spars.* ▲	*Acc.* ▲
GNNExp ($S = 1$)	0.65	1.00	0.92	0.61	0.69	1.00	0.96	0.79	0.90	1.00	0.94	0.52
GNNExp ($S = 2$)	0.59	2.00	0.85	0.54	0.51	2.00	0.92	0.78	0.85	2.00	0.91	0.40
GNNExp ($S = 3$)	0.56	3.00	0.79	0.51	0.46	3.00	0.88	0.79	0.83	3.00	0.87	0.34
GNNExp ($S = 4$)	0.58	4.00	0.72	0.48	0.42	4.00	0.84	0.79	0.83	4.00	0.83	0.31
GNNExp ($S = 5$)	0.57	5.00	0.66	0.46	0.40	5.00	0.80	0.79	0.81	5.00	0.81	0.27
GNNExp ($S = \mathrm{GT}$)	0.55	6.00	0.57	0.46	0.35	11.83	0.53	0.74	0.82	6.00	0.79	0.24
CF-GNNExplainer	**0.21**	2.09	0.90	**0.94**	**0.07**	1.47	0.94	**0.96**	**0.39**	2.39	0.99	**0.96**

CONTENTS

1. What is Explainability?
2. Explainable GNNs
3. Identifying Informative Substructures
   - Explainability
   - Expressivity
   - Causality
4. Conclusion

From Explainability to graph tasks

- Motivation: 왜 "중요한 Substructure"를 식별해야 하는가?
  - ✓ 많은 graph task에서 label과 속성은 종종 특정 substructure에 의해 결정됨
    - 분자 그래프: functional group, cycle
    - 소셜/바이오 네트워크: clique, motif

- Key components
  - ✓ **Explainability**: 임베딩에 무엇이 들어있나?를 substructure의 contribution으로 분해
  - ✓ **Expressivity**: MPNN이 못 보는 구조를 substructure로 보완
  - ✓ **Robustness/Causality**: Correlation vs. Causation

Pattern-based Explainable Graph Learning (IJCAI'25)

Explainable Graph Representation Learning via Graph Pattern Analysis

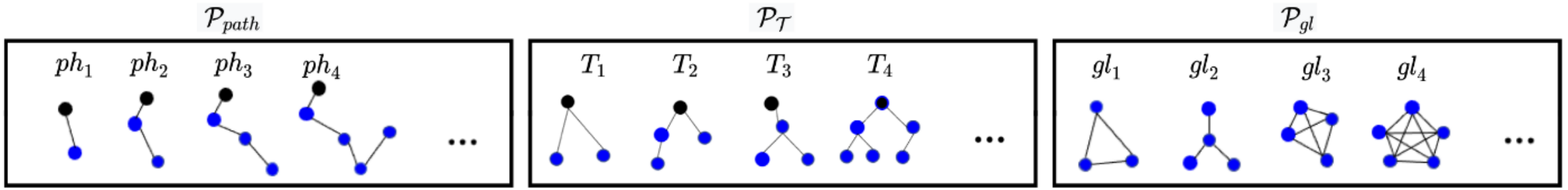
Xudong Wang[1] , **Ziheng Sun**[1,2] , **Chris Ding**[1] and **Jicong Fan**[1,2,*]

[1]School of Data Science, The Chinese University of Hong Kong, Shenzhen (CUHK-Shenzhen), China
[2]Shenzhen Research Institute of Big Data, Shenzhen, China
{xudongwang, zihengsun}@link.cuhk.edu.cn, {chrisding, fanjicong}@cuhk.edu.cn

- Motivation
  - ✓ Graph representation vector g 안에는 어떤 그래프 정보가 들어 있는가?

Pattern-based Explainable Graph Learning (IJCAI'25)

- Key Ideas
  - ✓ Subgraphs를 여러 종류의 그래프 패턴으로 그룹화
    - paths, trees, cycles, cliques, stars, ...
  - ✓ 각 패턴의 기여도를 가중치 λ로 명시적으로 학습
    - 기존 방식과 달리 그래프 내 모든 subgraphs를 분석하지 않아도 됨



Pattern-based Explainable Graph Learning (IJCAI'25)

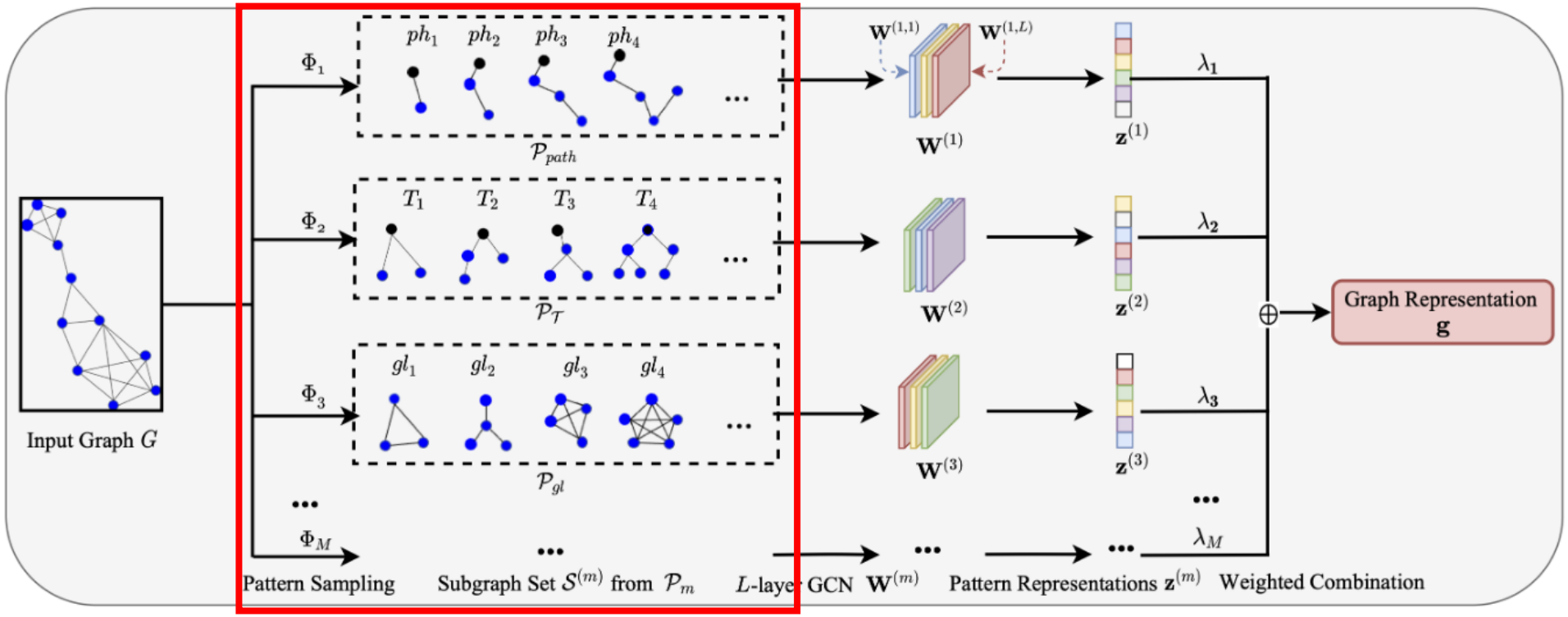
- Key Ideas
  - ✓ Subgraphs를 여러 종류의 그래프 패턴으로 그룹화
    - paths, trees, cycles, cliques, stars, ...

- **Paths:** S is a *path* if there exists a sequence of distinct vertices $v_1, \ldots, v_k \in V_S$ such that $E_S = ((v_i, v_{i+1}) : i = 1, \ldots, k-1)$.
- **Trees:** S is a *tree* if it is connected and contains no cycles, i.e., it is acyclic and $|E_S| = |V_S| - 1$.
- **Graphlets:** S is a *graphlet* if it is a small connected induced subgraph of G, typically consisting of 2 to 5 vertices.
- **Cycles:** S is a *cycle* if there exists a sequence of distinct vertices $v_1, \ldots, v_k \in V_S$ such that $E_S = ((v_i, v_{i+1}) : i = 1, \ldots, k-1) \cup ((v_k, v_1))$.
- **Cliques:** S is a *clique* if every two distinct vertices in V_S are adjacent, thus $E_S = ((v_i, v_j) : v_i, v_j \in V_S, i \neq j)$.
- **Wheels:** S is a *wheel* if it consists of a cycle with vertices $v_1, \ldots, v_{k-1}$ and an additional central vertex v_k such that v_k is connected to all vertices of the cycle.
- **Stars:** S is a *star* if it consists of one central vertex v_c and several leaf vertices $v_1, \ldots, v_{k-1}$, where each leaf vertex is only connected to v_c. Thus, $E_S = ((v_c, v_i) : i = 1, \ldots, k-1)$.

Pattern-based Explainable Graph Learning (IJCAI'25)

- Pattern Sampling Set
  - ✓ 그래프 내에서 각각의 패턴에 속하는 subgraphs를 샘플링

$$\mathcal{S} := \{S_1, S_2, \ldots, S_q, \ldots, S_Q\}, \quad \text{where} \quad S_q \in \mathcal{P}, \ \forall\, q \in [Q].$$

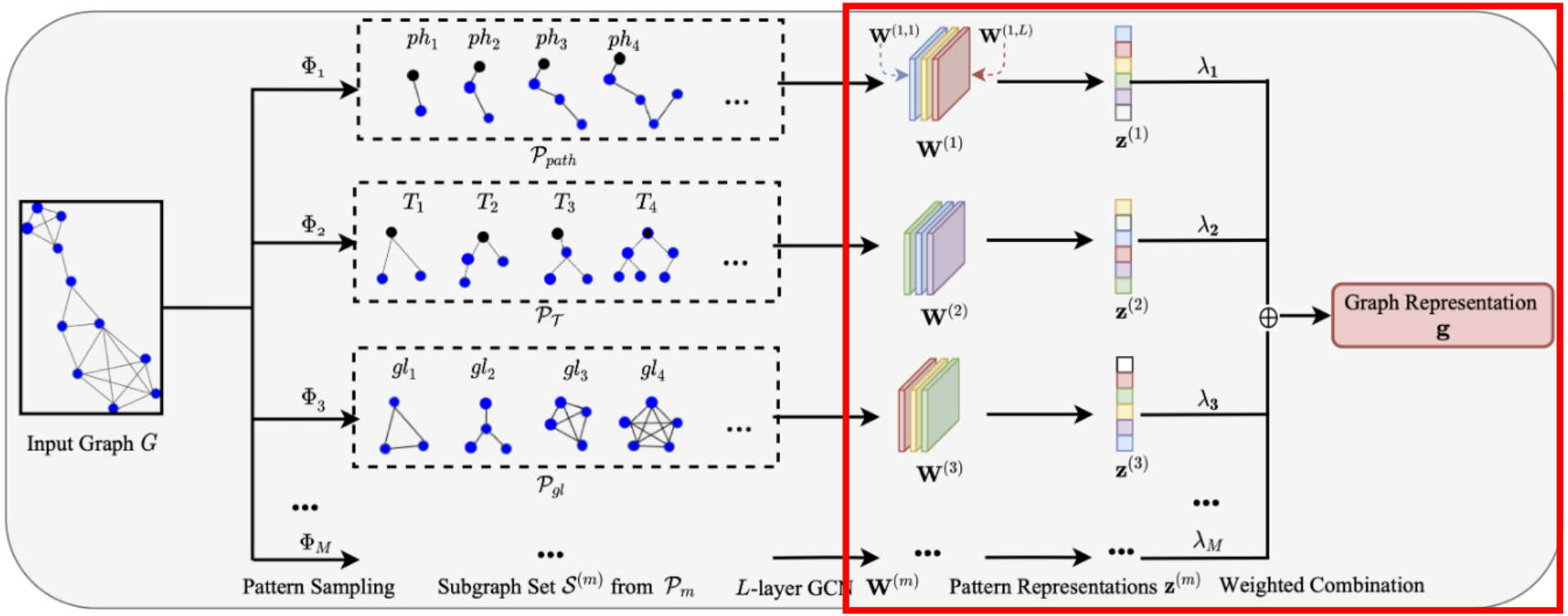


Pattern-based Explainable Graph Learning (IJCAI'25)

- Pattern Representation
  - ✓ GNN으로 패턴별 representation 학습
- 가중치 자체가 Explanation

$$\boldsymbol{g} = \sum_{m=1}^{M} \lambda_m \boldsymbol{z}^{(m)}, \quad \text{with}$$

$$\boldsymbol{z}^{(m)} = \frac{1}{|\mathcal{S}^{(m)}|} \sum_{S \in \mathcal{S}^{(m)}} F(\boldsymbol{A}_S, \boldsymbol{X}_S; \mathcal{W}^{(m)}), \quad \forall\, m \in [M].$$



Pattern-based Explainable Graph Learning (IJCAI'25)

- Representation-level explanation

Pattern	MUTAG	PROTEINS	DD	NCI1	COLLAB	IMDB-B	REDDIT-B	REDDIT-M5K
paths	0.095 ± 0.014	$\mathbf{0.550 \pm 0.070}$	0.093 ± 0.012	0.022 ± 0.002	$\mathbf{0.587 \pm 0.065}$	0.145 ± 0.018	0.131 ± 0.027	0.027 ± 0.003
trees	0.046 ± 0.005	0.074 ± 0.009	0.054 ± 0.006	0.063 ± 0.008	0.105 ± 0.013	0.022 ± 0.003	0.055 ± 0.007	0.025 ± 0.003
graphlets	0.062 ± 0.008	0.081 ± 0.011	0.125 ± 0.015	0.101 ± 0.013	0.063 ± 0.008	0.084 ± 0.011	0.026 ± 0.003	0.054 ± 0.007
cycles	$\mathbf{0.654 \pm 0.085}$	0.099 ± 0.013	0.094 ± 0.012	0.176 ± 0.022	0.022 ± 0.003	0.123 ± 0.016	0.039 ± 0.005	0.037 ± 0.005
cliques	0.082 ± 0.011	0.098 ± 0.012	$\mathbf{0.572 \pm 0.073}$	$\mathbf{0.574 \pm 0.075}$	0.134 ± 0.017	$\mathbf{0.453 \pm 0.054}$	0.279 ± 0.069	0.256 ± 0.067
wheels	0.026 ± 0.003	0.039 ± 0.005	0.051 ± 0.007	0.012 ± 0.002	0.068 ± 0.009	0.037 ± 0.004	0.036 ± 0.005	0.023 ± 0.003
stars	0.035 ± 0.005	0.056 ± 0.007	0.011 ± 0.002	0.052 ± 0.007	0.021 ± 0.003	0.136 ± 0.017	$\mathbf{0.447 \pm 0.006}$	$\mathbf{0.578 \pm 0.033}$

Table 2: The learned $\boldsymbol{\lambda}$ of PXGL-GNN (supervised). The largest value is **bold** and the second largest value is blue.

Method	MUTAG	PROTEINS	DD	NCI1	COLLAB	IMDB-B	REDDIT-B	REDDIT-M5K
GIN	84.53 ± 2.38	73.38 ± 2.16	76.38 ± 1.58	73.36 ± 1.78	75.83 ± 1.29	72.52 ± 1.62	83.27 ± 1.30	52.48 ± 1.57
DiffPool	86.72 ± 1.95	76.07 ± 1.62	77.42 ± 2.14	75.42 ± 2.16	78.77 ± 1.36	73.55 ± 2.14	84.16 ± 1.28	51.39 ± 1.48
DGCNN	84.29 ± 1.16	75.53 ± 2.14	76.57 ± 1.09	74.81 ± 1.53	77.59 ± 2.24	72.19 ± 1.97	86.33 ± 2.29	53.18 ± 2.41
GRAPHSAGE	86.35 ± 1.31	74.21 ± 1.85	79.24 ± 2.25	77.93 ± 2.04	76.37 ± 2.11	73.86 ± 2.17	85.59 ± 1.92	51.65 ± 2.55
SubGNN	87.52 ± 2.37	76.38 ± 1.57	82.51 ± 1.67	82.58 ± 1.79	81.26 ± 1.53	71.58 ± 1.20	88.47 ± 1.83	53.27 ± 1.93
SAN	92.65 ± 1.53	75.62 ± 2.39	81.36 ± 2.10	83.07 ± 1.54	82.73 ± 1.92	75.27 ± 1.43	90.38 ± 1.54	55.49 ± 1.75
SAGNN	93.24 ± 2.51	75.61 ± 2.28	84.12 ± 1.73	81.29 ± 1.22	79.94 ± 1.83	74.53 ± 2.57	89.57 ± 2.13	54.11 ± 1.22
ICL	91.34 ± 2.19	75.44 ± 1.26	82.77 ± 1.42	83.45 ± 1.78	81.45 ± 1.21	73.29 ± 1.46	90.13 ± 1.40	56.21 ± 1.35
S2GAE	89.27 ± 1.53	76.47 ± 1.12	84.30 ± 1.77	82.37 ± 2.24	82.35 ± 2.34	75.77 ± 1.72	90.21 ± 1.52	54.53 ± 2.17
PXGL-GNN	$\mathbf{94.87 \pm 2.26}$	$\mathbf{78.23 \pm 2.46}$	$\mathbf{86.54 \pm 1.95}$	$\mathbf{85.78 \pm 2.07}$	$\mathbf{83.96 \pm 1.59}$	$\mathbf{77.35 \pm 2.32}$	$\mathbf{91.84 \pm 1.69}$	$\mathbf{57.36 \pm 2.14}$

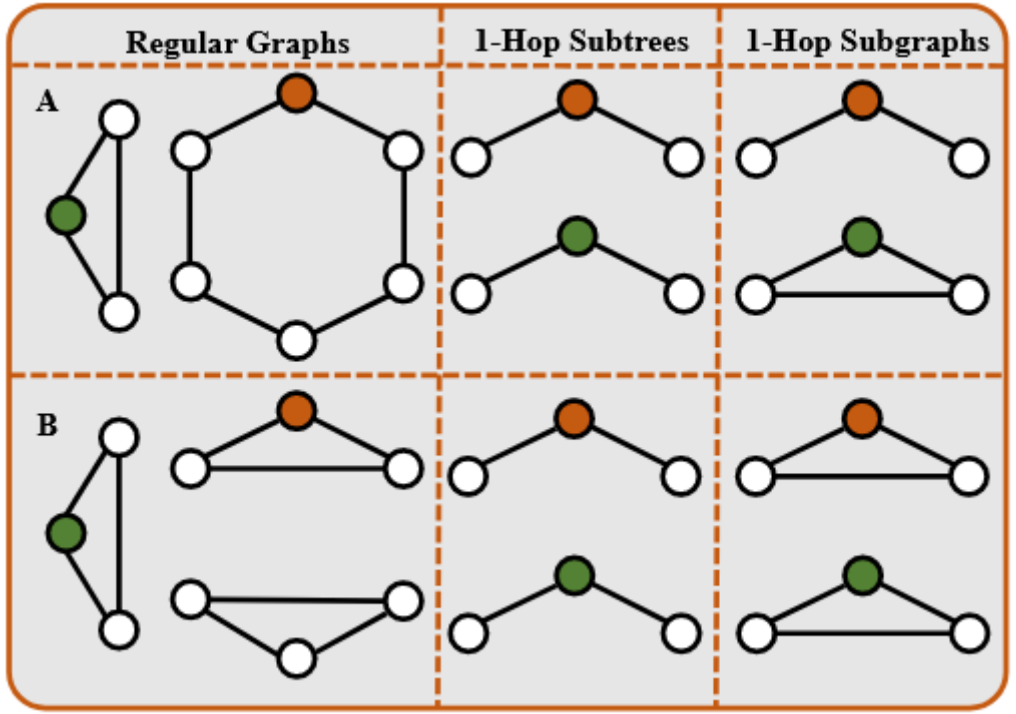
Table 3: Accuracy (%) of Graph Classification. The best accuracy is **bold** and the second best is blue.

Pattern-based Explainable Graph Learning (IJCAI'25)

- PXGL-GNN은 임베딩을 패턴 축으로 설명
- 그러나 근본적으로:
  - ✓ 모델이 cycle/clique 같은 substructure를 표현적으로 잘 인식해야 λ 해석의 설득력이 커짐
  - ✓ 모든 subgraph를 열거하는 것은 비용이 매우 큼

Expressivity

- "그래프를 전부 보지 말고, 구분에 필요한 subgraph만 찾자"
  - ✓ 기존 subgraph GNN은 표현력이 좋지만 모든 rooted subgraph를 나열하는 것은 비용이 큼
  - ✓ 구분에 필요한 건 모든 subgraph가 아니라 소수의 discriminative subgraph일 수 있음



[1] MAG-GNN: Reinforcement Learning Boosted Graph Neural Network, NIPS'23

Agent-based GNN (ICLR'23)

AGENT-BASED GRAPH NEURAL NETWORKS

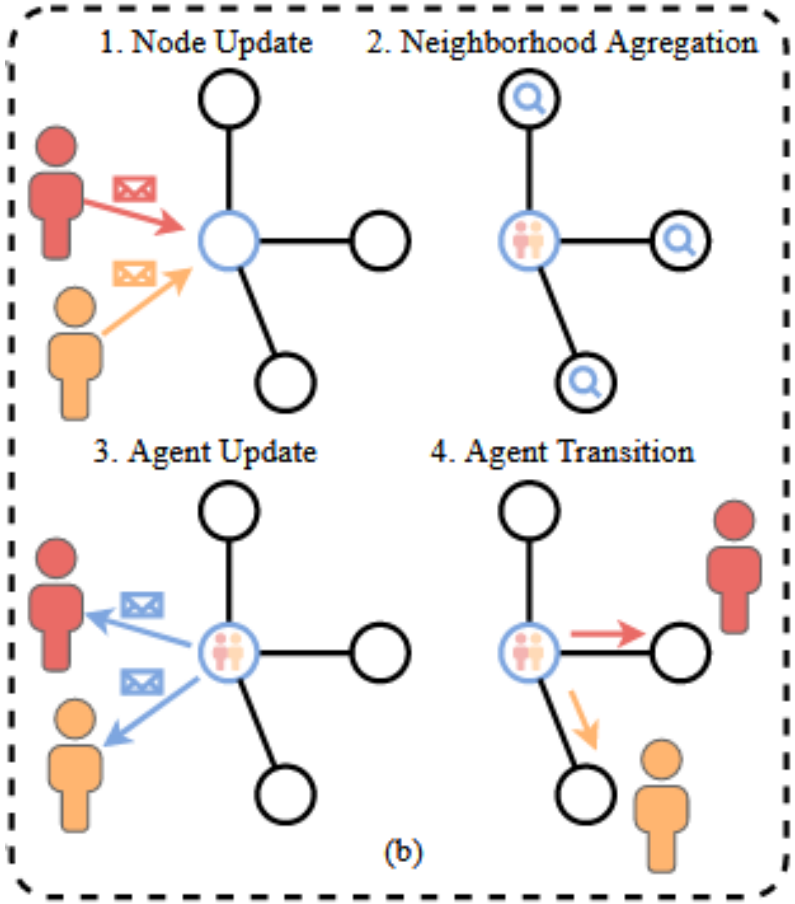
Karolis Martinkus[1], **Pál András Papp**[2], **Benedikt Schesch**[1], **Roger Wattenhofer**[1]
[1]ETH Zurich [2]Computing Systems Lab, Huawei Zurich Research Center

Cited: 33

- Motivation: "전체를 다 볼 필요가 있을까?"
  - ✓ 기존 GNN은 모든 노드가 L번의 message passing + pooling
  - ✓ 하지만 graph label은 일부 substructure가 좌우
  - ✓ sublinear한 분류 가능성

Agent-based GNN (ICLR'23)

- Graph-level 예측은 그래프 위를 지능적으로 여러 번 병렬 탐색함으로써 수행 가능함
  - ✓ 각 agent는 고유 식별 ID를 가지며, 그래프 위에 랜덤하게 초기화됨
  - ✓ 4단계 steps
    1. Node update
    2. Neighborhood aggregation
    3. Agent update
    4. Agent transition

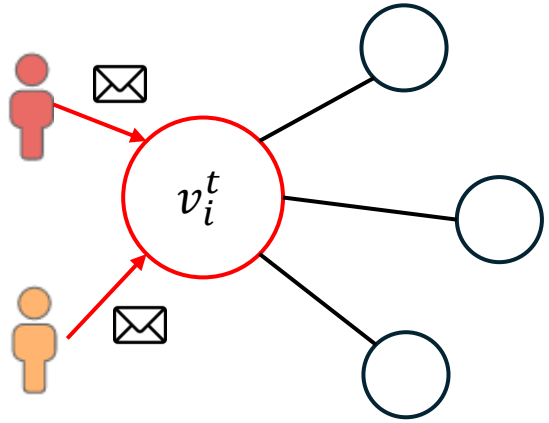


Agent-based GNN (ICLR'23)

① Node update

- 해당 노드에 올라온 agent들의 정보를 집계해 노드 상태를 업데이트

$$v_i^t = f_v\left(v_i^{t-1}, \sum_{a_j^{t-1} \in A(v_i)} \phi_{a \to v}(a_j^{t-1})\right) \quad \textbf{if } |A(v_i)| > 0 \textbf{ else } v_i^{t-1}.$$

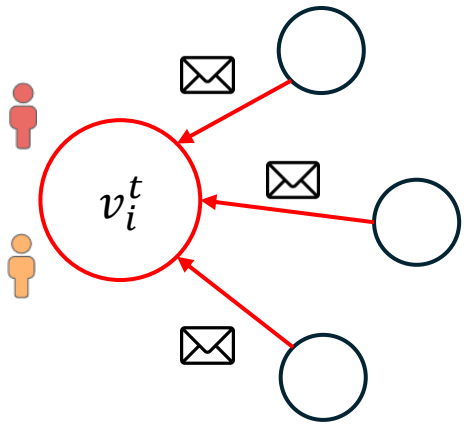


Agent-based GNN (ICLR'23)

② Neighborhood aggregation

- 이웃 노드 임베딩을 현재 timestep의 노드 상태에 통합

$$v_i^t = f_n\left(v_i^t, \sum_{v_j^t \in N(v_i)} \phi_{N(v)\to v}(v_j^t)\right) \quad \textbf{if } |A(v_i)| > 0 \textbf{ else } v_i^t.$$

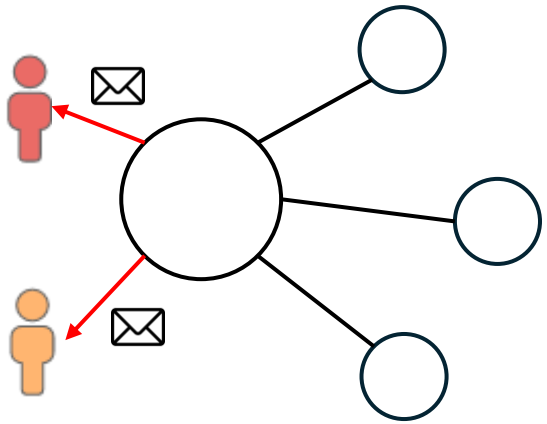


Agent-based GNN (ICLR'23)

③ Agent update

- 업데이트된 노드 임베딩을 사용하여 에이전트 임베딩을 업데이트
  - ✓ 이로써 에이전트는 현재 이웃의 상태와 같은 노드에 있는 다른 에이전트의 상태를 알게 됨

$$a_i^t = f_a\left(a_i^{t-1}, v_{V(a_i)}^t\right).$$



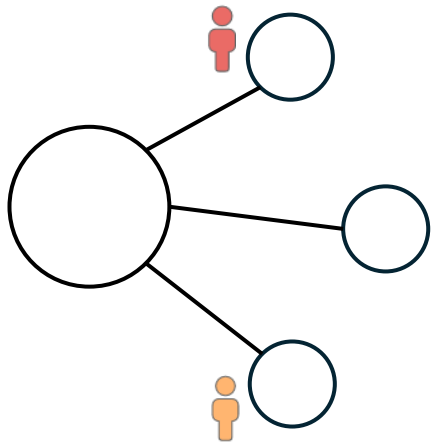
Agent-based GNN (ICLR'23)

④ Agent transition

- 에이전트가 현재 노드와 그 이웃을 모두 포함하는 잠재적 다음 위치로 전이
  - ✓ 전이 확률 logit은 dot-product attention으로 계산됨
  - ✓ Gumbel softmax estimator를 사용해 다음 위치를 샘플링

$$z_{a_i \to v_j} = f_p\left(a_i^t, v_j^t\right) \quad \text{for} \quad v_j^t \in N^t(a_i),$$

$$V(a_i) \leftarrow \textbf{GumbelSoftmax}\left(\left\{z_{a_i \to v_j} \quad \text{for} \quad v_j^t \in N^t(a_i)\right\}\right).$$



- 최종 graph-level 예측을 수행할 때는 readout (MLP) 적용 후 agent 임베딩을 풀링

Agent-based GNN (ICLR'23)

Theoretical Analysis

- AgentNet이 1-WL test를 능가하는 것은 물론, 2-WL test로도 구별할 수 없는 구조를 인식하며, subgraph 빈도에 따라 효율적으로 그래프를 분류할 수 있음을 증명
  - ✓ 단일 agent 분석: agent 하나가 이웃을 얼마나 깊이 이해할 수 있는가?
    - Lemma 1: AgentNet은 r-hop 이웃에 대해 반복적으로 depth를 심화하는 DFS (IDDFS)를 학습하여 $O(r \cdot |N^r(v)|)$ 단계 만에 모든 노드를 방문할 수 있음
    - Theorem 2: Agent가 충분히 많은 단계를 거치면, non-isomorphic인 모든 r-hop 이웃에 대해 서로 다른 최종 임베딩을 계산할 수 있음
      - 이는 agent가 이웃 내의 모든 node, edge, 심지어 feature까지 완벽하게 식별할 수 있음을 의미
  - ✓ 다중 agent 분석: 병렬성과 표현력의 관계
    - Agent는 고유 ID를 통해 다른 agent의 행동과 무관하게 독립적으로 작동할 수 있음
      - 즉, agent 추가가 표현력을 감소시키지 않음

Agent-based GNN (ICLR'23)

Theoretical Analysis

- AgentNet이 1-WL test를 능가하는 것은 물론, 2-WL test로도 구별할 수 없는 그래프 구조를 이산하며, subgraph 빈도에 따라 효

  ✓ 단

  ▪

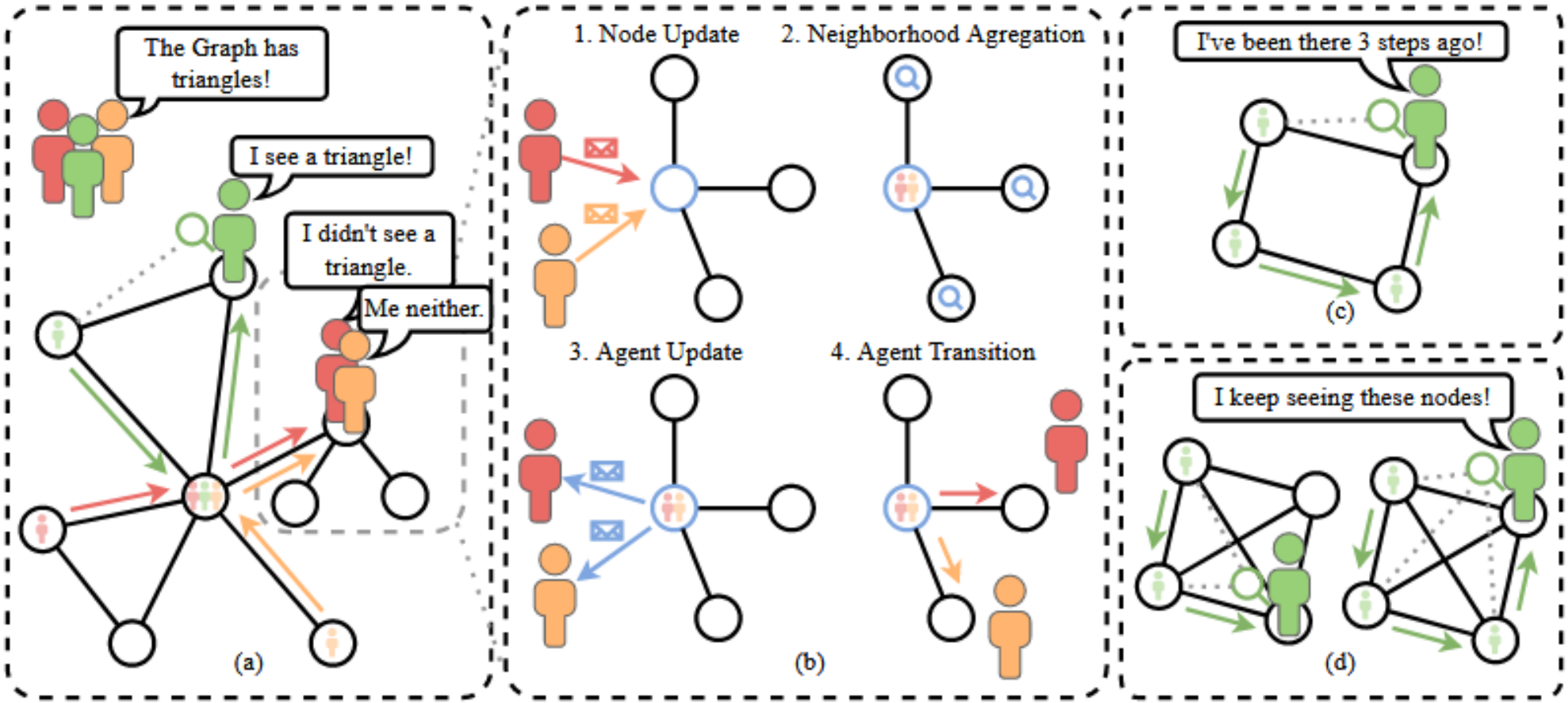
  ▪

  ✓ 다

  ▪

에 모든 노

할 수 있음



Agent-based GNN (ICLR'23)

Theoretical Analysis

- k-WL test
  - ✓ 두 그래프가 서로 동형인지 여부를 판별하는 데 사용하는 일련의 반복적인 정제 알고리즘
    - 1-WL test: Node Color Refinement
      - GNN, 특히 message passing 기반의 GNN의 표현력과 연결됨
      - cycle, clique, 또는 일부 복잡한 substructure의 유무를 구별하지 못하는 경우가 많으며, 이것이 대부분의 표준 GNN(예: GCN, GAT, GIN)의 이론적 표현력의 상한선
    - k-WL test: k-Tuple Color Refinement
      - k개의 노드로 구성된 tuple을 기본 단위로 사용
      - 2-WL test는 node pair를 기본 단위로 하며, 1-WL test로 구별할 수 없는 일부 구조를 구별할 수 있음
  - ✓ k가 충분히 크다면, k-WL test는 임의의 두 그래프가 동형인지 아닌지를 완벽히 판별할 수 있음

Agent-based GNN (ICLR'23)

Theoretical Analysis

- **Sublinear 계산의 가능성**
  - ✓ AgentNet은 graph classification이 항상 전체 그래프를 필요로 하지 않는다는 아이디어에 기반함

Theorem 9

Graph class를 결정하는 특정 subgraph가 한 그래프에서 다른 그래프보다 충분히 더 자주 나타나기만 한다면, agent가 그래프의 일부만 탐색하더라도 O(1)개의 적은 agent로도 두 그래프를 높은 확률로 구별할 수 있음

Agent-based GNN (ICLR'23)

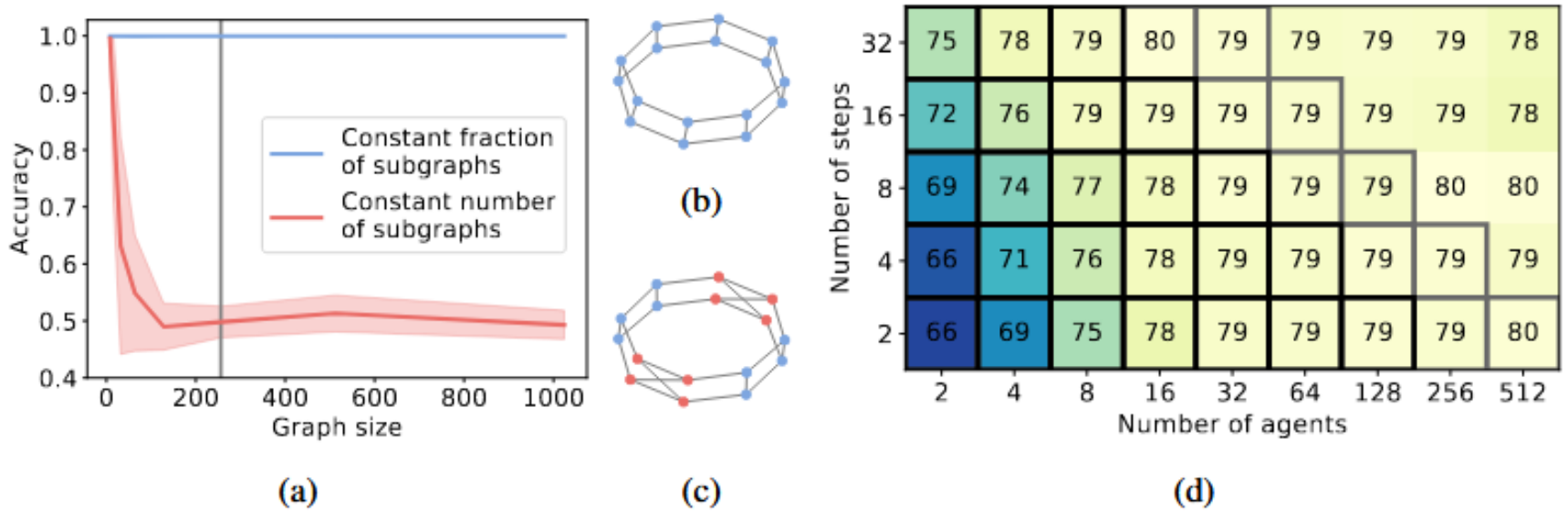
Theoretical Analysis

- 지능적인 Agent Transition
  - ✓ AgentNet의 성능은 agent가 다음 노드를 랜덤하게 선택하지 않고 지능적으로 결정하는 능력에서 나옴
    - 다음 노드를 선택하는 함수는 Dot-product attention으로 구현됨
      - Agent는 자신의 상태(Query)와 이웃 노드의 상태(Key)를 비교하여 각 이웃으로의 선호도(logit)을 계산함
    - 이 logit은 Gumbel softmax를 통해 미분 가능한 방식으로 다음 노드를 샘플링하는 데 사용됨
      - Agent의 이동 전략 전체를 gradient descent로 학습

Agent-based GNN (ICLR'23)

Experiments: Expressiveness on Synthetic Datasets

- 정의된 substructure가 그래프에 퍼져있을 때 AgentNet이 두 그래프를 실제로 구별할 수 있는지 실험
  - ✓ 이 경우 AgentNet은 그래프의 일부분만 관찰하더라도 성공적으로 그래프를 구별함
  - ✓ 16개의 agent, 16 step으로 실험
  - ✓ AgentNet은 $\frac{n}{8}$ 노드만 방문했을 때 GIN보다 우수한 성능, $\frac{n}{16}$ 노드 방문 시 동일한 성능



Agent-based GNN (ICLR'23)

Experiments: Efficency on Real-world Datasets

- 최대 1-WL 표현력을 가진 GIN과 더 표현력이 높은 GNN 아키텍처와 비교
  - ✓ 고차 GNN (PPGN, 1-2-3 GNN)은 대형 그래프에서 OOM으로 실패
  - ✓ DD dataset에서 AgentNet의 노드 방문 횟수 ($k \cdot l$)가 전체 노드 수보다 적은 설정에서도 GIN과 비슷하거나 그 이상
  - ✓ Higher-order GNN은 24GB GPU에서는 대형 그래프를 훈련조차 못 함
    - 그래서 ESAN, DropGIN을 훈련할 때 그래프의 5%만 활용할 수밖에 없었음

Model	Complexity	MUTAG	PTC	PROTEINS	IMDB-B	IMDB-M	DD	RDT-B
GIN [75]	$O(n \cdot \ell)$	89.4 +5.6	64.6 +7.0	76.2 +2.8	75.1 +5.1	52.3 +2.8	76.9 +3.7	92.4 +2.5
DropGIN [59]	$O(r \cdot n \cdot \ell)$	90.4 +7.0	66.3 +8.6	76.3 +6.1	75.7 +4.2	51.4 +2.8	76.4 +3.4	89.9 +1.7
ESAN [8]*	$O(r \cdot n \cdot \ell)$	91.1 +7.0	**69.2 +6.5**	77.1 +4.6	**77.1 +2.6**	**53.5 +3.4**	**81.2 +2.3**	93.3 +1.3
1-2-3 GNN [53]†	$O(n^4 \cdot \ell)$	88.8 +7.0	64.0 +6.0	76.8 +3.7	73.6 +2.2	51.1 +3.8	OOM	OOM
PPGN [51]*	$O(n^3 \cdot \ell)$	90.6 +8.7	66.2 +6.5	**77.2 +4.7**	73 +5.8	50.5 +3.6	OOM	OOM
CRaWl [67]	$O((n + k \cdot m) \cdot \ell)$	90.4 +7.1	68.0 +6.5	76.2 +3.7	73.4 +2.1	47.8 +3.9	78.3 +5.5	92.8 +2.2
1-AgentNet	$O(\ell)$	89.4 +10.9	66.6 +7.6	75.1 +3.4	74.9 +3.9	52.3 +3.9	67.4 +3.0	77.9 +3.0
AgentNet	$O(k \cdot \ell)$	**93.6 +8.6**	67.4 +5.9	76.7 +3.2	75.2 +4.6	52.2 +3.8	80.1 +2.7	**94.2 +1.2**
Rank		1st	3rd	4th	3rd	3rd	2nd	1st

Remind PXGL-GNN …

- Patterns can be correlated, not causal

Pattern	MUTAG	PROTEINS	DD	NCI1	COLLAB	IMDB-B	REDDIT-B	REDDIT-M5K
paths	0.095 ± 0.014	$\mathbf{0.550 \pm 0.070}$	0.093 ± 0.012	0.022 ± 0.002	$\mathbf{0.587 \pm 0.065}$	0.145 ± 0.018	0.131 ± 0.027	0.027 ± 0.003
trees	0.046 ± 0.005	0.074 ± 0.009	0.054 ± 0.006	0.063 ± 0.008	0.105 ± 0.013	0.022 ± 0.003	0.055 ± 0.007	0.025 ± 0.003
graphlets	0.062 ± 0.008	0.081 ± 0.011	0.125 ± 0.015	0.101 ± 0.013	0.063 ± 0.008	0.084 ± 0.011	0.026 ± 0.003	0.054 ± 0.007
cycles	$\mathbf{0.654 \pm 0.085}$	0.099 ± 0.013	0.094 ± 0.012	0.176 ± 0.022	0.022 ± 0.003	0.123 ± 0.016	0.039 ± 0.005	0.037 ± 0.005
cliques	0.082 ± 0.011	0.098 ± 0.012	$\mathbf{0.572 \pm 0.073}$	$\mathbf{0.574 \pm 0.075}$	0.134 ± 0.017	$\mathbf{0.453 \pm 0.054}$	0.279 ± 0.069	0.256 ± 0.067
wheels	0.026 ± 0.003	0.039 ± 0.005	0.051 ± 0.007	0.012 ± 0.002	0.068 ± 0.009	0.037 ± 0.004	0.036 ± 0.005	0.023 ± 0.003
stars	0.035 ± 0.005	0.056 ± 0.007	0.011 ± 0.002	0.052 ± 0.007	0.021 ± 0.003	0.136 ± 0.017	$\mathbf{0.447 \pm 0.006}$	$\mathbf{0.578 \pm 0.033}$

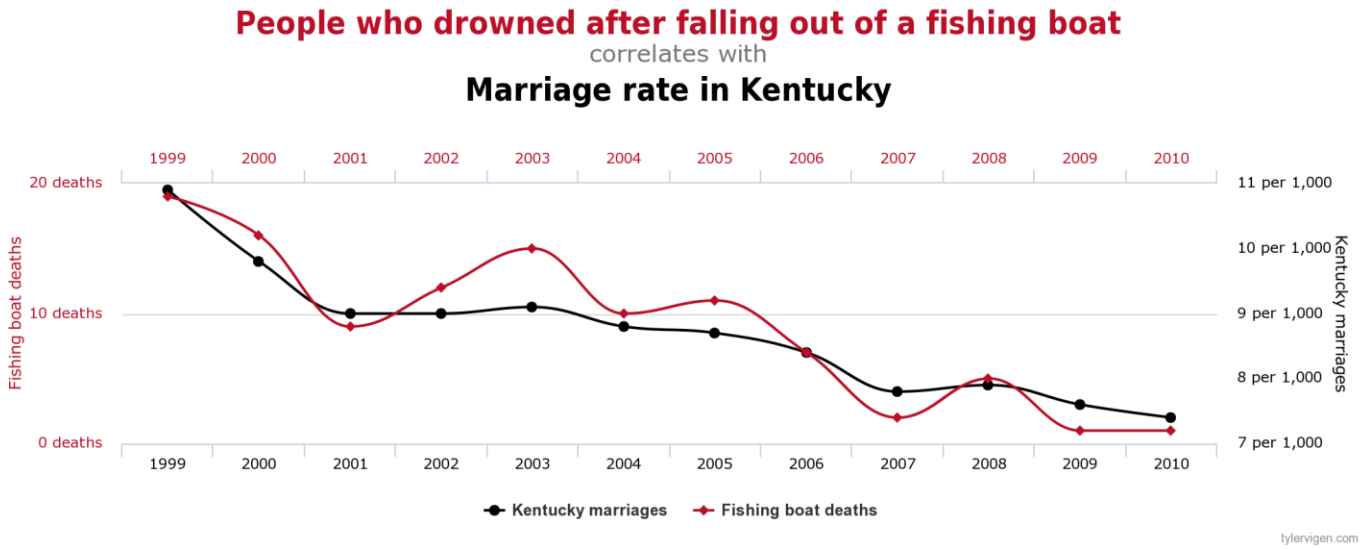
Table 2: The learned $\boldsymbol{\lambda}$ of PXGL-GNN (supervised). The largest value is **bold** and the second largest value is blue.

Method	MUTAG	PROTEINS	DD	NCI1	COLLAB	IMDB-B	REDDIT-B	REDDIT-M5K
GIN	84.53 ± 2.38	73.38 ± 2.16	76.38 ± 1.58	73.36 ± 1.78	75.83 ± 1.29	72.52 ± 1.62	83.27 ± 1.30	52.48 ± 1.57
DiffPool	86.72 ± 1.95	76.07 ± 1.62	77.42 ± 2.14	75.42 ± 2.16	78.77 ± 1.36	73.55 ± 2.14	84.16 ± 1.28	51.39 ± 1.48
DGCNN	84.29 ± 1.16	75.53 ± 2.14	76.57 ± 1.09	74.81 ± 1.53	77.59 ± 2.24	72.19 ± 1.97	86.33 ± 2.29	53.18 ± 2.41
GRAPHSAGE	86.35 ± 1.31	74.21 ± 1.85	79.24 ± 2.25	77.93 ± 2.04	76.37 ± 2.11	73.86 ± 2.17	85.59 ± 1.92	51.65 ± 2.55
SubGNN	87.52 ± 2.37	76.38 ± 1.57	82.51 ± 1.67	82.58 ± 1.79	81.26 ± 1.53	71.58 ± 1.20	88.47 ± 1.83	53.27 ± 1.93
SAN	92.65 ± 1.53	75.62 ± 2.39	81.36 ± 2.10	83.07 ± 1.54	82.73 ± 1.92	75.27 ± 1.43	90.38 ± 1.54	55.49 ± 1.75
SAGNN	93.24 ± 2.51	75.61 ± 2.28	84.12 ± 1.73	81.29 ± 1.22	79.94 ± 1.83	74.53 ± 2.57	89.57 ± 2.13	54.11 ± 1.22
ICL	91.34 ± 2.19	75.44 ± 1.26	82.77 ± 1.42	83.45 ± 1.78	81.45 ± 1.21	73.29 ± 1.46	90.13 ± 1.40	56.21 ± 1.35
S2GAE	89.27 ± 1.53	76.47 ± 1.12	84.30 ± 1.77	82.37 ± 2.24	82.35 ± 2.34	75.77 ± 1.72	90.21 ± 1.52	54.53 ± 2.17
PXGL-GNN	$\mathbf{94.87 \pm 2.26}$	$\mathbf{78.23 \pm 2.46}$	$\mathbf{86.54 \pm 1.95}$	$\mathbf{85.78 \pm 2.07}$	$\mathbf{83.96 \pm 1.59}$	$\mathbf{77.35 \pm 2.32}$	$\mathbf{91.84 \pm 1.69}$	$\mathbf{57.36 \pm 2.14}$

Table 3: Accuracy (%) of Graph Classification. The best accuracy is **bold** and the second best is blue.

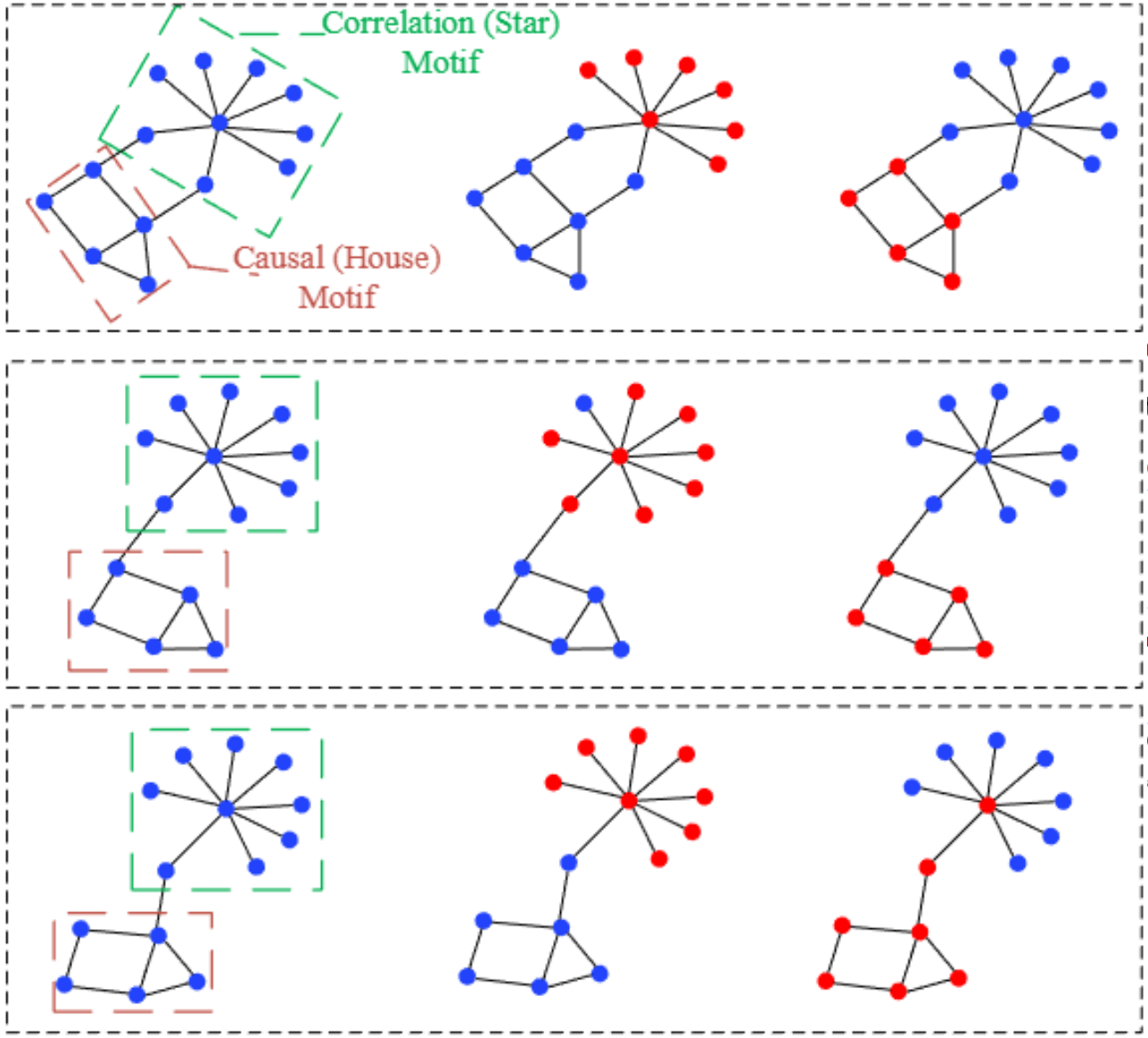
Spurious Correlation

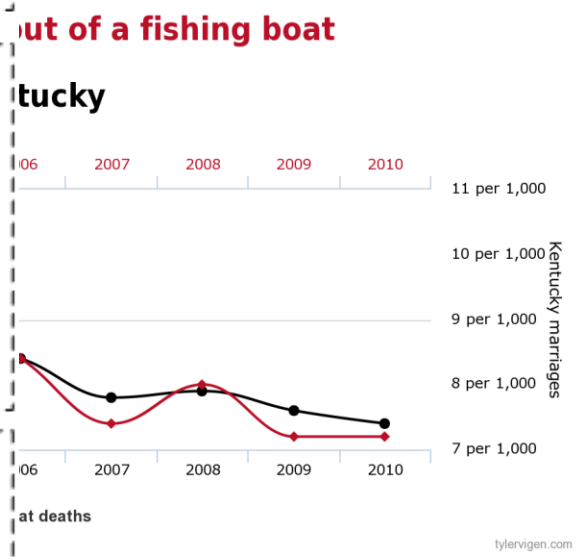
- Focus on “shortcuts”
  - ✓ Pentagon Pizza Index (Pizza Meter)
  - ✓ Marriage rate in Kentucky



Spurious Correlation

- Focus on “shortcuts”
  - ✓ Pentagon Pizza Inde
  - ✓ Marriage rate in Kent





Information-based Causal Learning (AAAI'24)

A Twist for Graph Classification: Optimizing Causal Information Flow in Graph Neural Networks

Zhe Zhao[13], Pengkun Wang[12*], Haibin Wen[4], Yudong Zhang[1], Zhengyang Zhou[12], Yang Wang [125*]

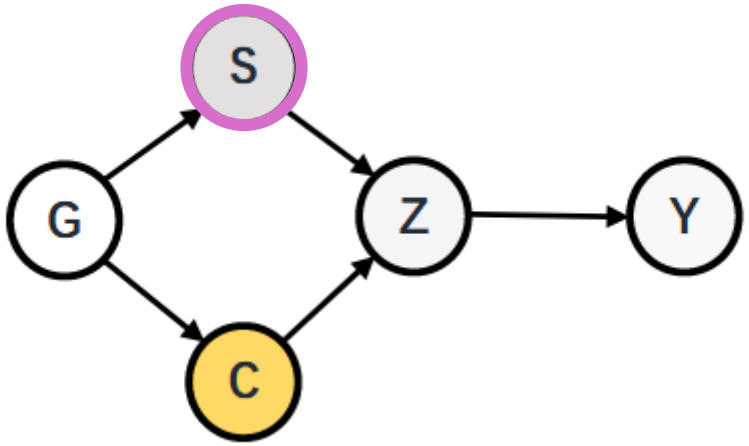
[1]University of Science and Technology of China, Hefei 230026, China
[2]Suzhou Institute for Advanced Research, University of Science and Technology of China, Suzhou 215123, China
[3]City University of Hong Kong
[4]Shaoguan University
[5]Key Laboratory of Precision and Intelligent Chemistry, USTC
{zz4543, zyd2020}@mail.ustc.edu.cn, {pengkun, zzy0929, angyan}@ustc.edu.cn, haibin65535@gmail.com

Cited: 20

- Correlation is not a "causation"
  - ✓ 기존 mutual information 최대화 방식은 $I(Z;Y)$ 를 키우는 방향 (correlation maximization)
  - ✓ 하지만 correlation에는 casual feature + non-causal feature가 섞여있음 -> out-of-distribution (OOD)에서 깨짐

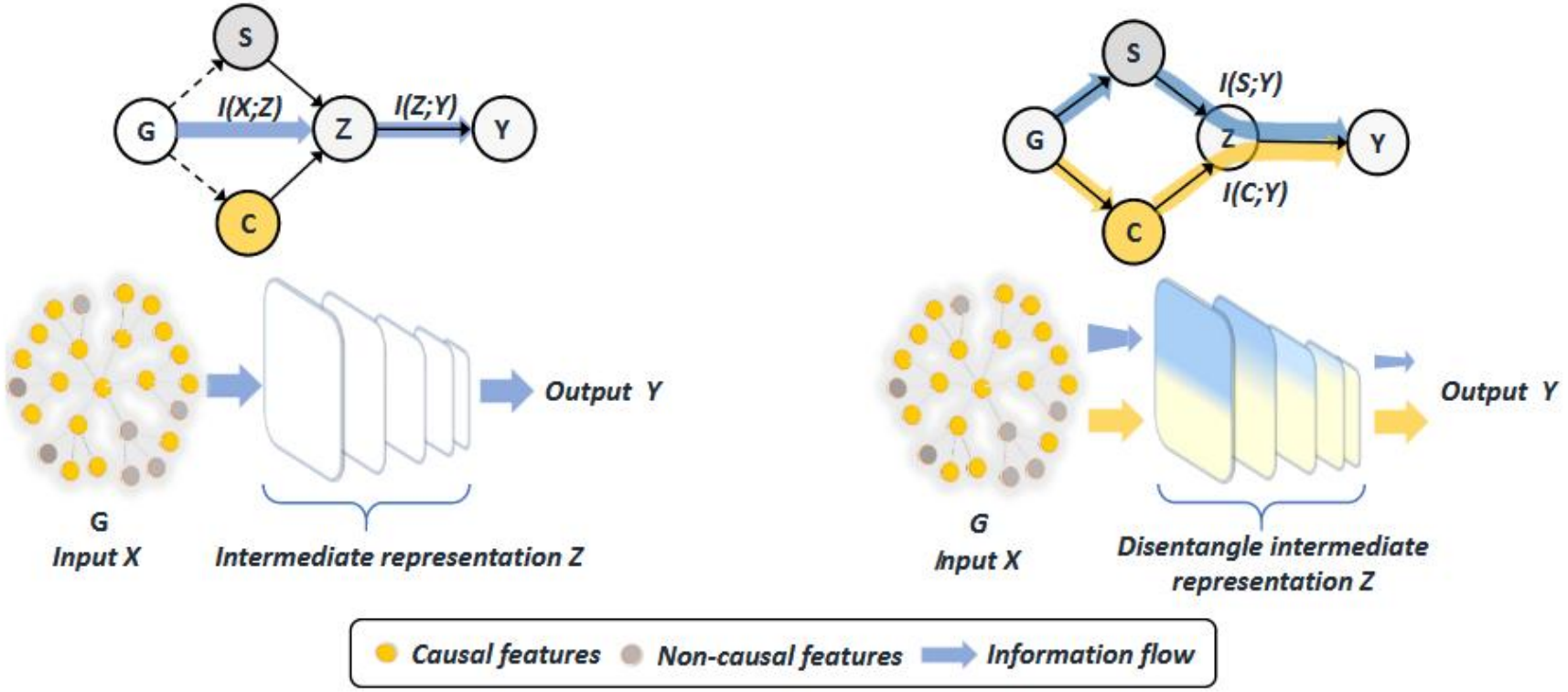
Information-based Causal Learning (AAAI'24)

- Structural Causal Model (SCM)
  - ✓ $C \leftarrow G \rightarrow S$: C는 causal features, S는 non-causal features이며, input G에는 둘이 공존함
  - ✓ $Z \rightarrow Y$: 최종 목표는 graph representation을 통해 label을 예측하는 것
  - ✓ $C \rightarrow Z \leftarrow S$: Z는 주어진 graphical data G의 representation으로, GNN 모델은 causal과 non-causal features를 동시에 함께 사용하여 학습함



Information-based Causal Learning (AAAI'24)

- Causal node/edge와 non-causal node/edge를 각각 학습하여 3개의 objective를 동시에 최적화하도록 함



Information-based Causal Learning (AAAI'24)

- Mutual information chain rule을 이용해 causal/non-causal 관계를 구분

$$I(Z;Y) = I(C,S;Y) = I(C;Y) + I(S;Y|C)$$

  - ✓ $I(C;Y)$는 true information과 causal dependence를 의미
  - ✓ $I(S;Y|C)$는 noise와 C로는 설명될 수 없는 non-causal dependencies를 의미

- 단순히 $I(Z;Y)$를 최적화하는 것은 위 두 term이 함께 증가할 것이고, 이는 non-causal dependencies 때문에 일반화 능력을 저해함

- 따라서, objective를 아래 식으로 대체함

$$\max I(Z;Y) \;\&\; \max I(C;Y) \;\&\; \min I(S;Y|C)$$

Information-based Causal Learning (AAAI'24)

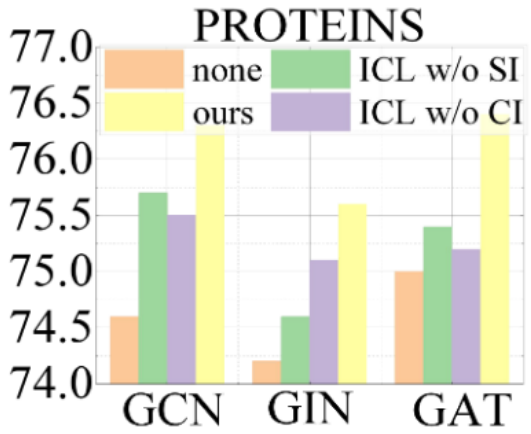
- Test accuracy of classification on TUDataset

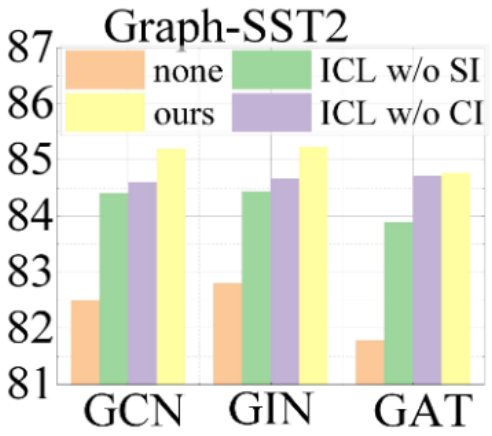
Method	MUTAG	NCI1	PROTEINS	COLLAB	IMDB-B	IMDB-M	AVG
DiffPool	85.61 ± 6.22	75.06 ± 3.66	76.25 ± 4.21	79.24 ± 1.66	74.47 ± 3.84	49.20 ± 3.10	76.38
SortPool	86.17 ± 7.53	79.00 ± 1.68	75.48 ± 1.62	77.84 ± 1.22	73.00 ± 3.50	49.53 ± 2.29	76.49
AGNN	79.77 ± 8.54	79.96 ± 2.37	75.66 ± 3.94	81.10 ± 2.39	73.10 ± 4.07	49.73 ± 3.72	76.15
GCN	88.20 ± 7.33	82.97 ± 2.34	75.65 ± 3.24	81.72 ± 1.64	73.89 ± 5.74	51.53 ± 3.28	78.00
GCN + CAL	88.89 ± 7.16	83.16 ± 1.73	73.32 ± 2.20	82.24 ± 1.62	74.00 ± 5.68	51.7 ± 3.37	79.05
GCN + ICL	88.33 ± 6.89	83.21 ± 2.17	74.93 ± 2.88	82.68 ± 1.90	74.70 ± 5.21	51.27 ± 3.02	79.22
GIN	89.42 ± 7.40	82.71 ± 1.52	76.21 ± 3.83	82.08 ± 1.51	73.40 ± 3.78	51.53 ± 2.97	78.35
GIN + CAL	87.81 ± 10.51	82.73 ± 2.24	73.22 ± 3.46	82.66 ± 1.93	73.60 ± 5.70	51.47 ± 2.77	78.31
GIN + ICL	88.39 ± 8.80	83.36 ± 2.22	75.02 ± 3.51	82.68 ± 1.06	74.50 ± 4.09	52.00 ± 4.18	79.35
GAT	88.58 ± 7.54	82.11 ± 1.43	75.96 ± 3.26	81.42 ± 1.41	72.70 ± 4.37	50.60 ± 3.75	77.88
GAT + CAL	88.83 ± 6.82	83.36 ± 0.85	74.40 ± 4.14	81.86 ± 1.42	71.90 ± 5.20	50.07 ± 2.84	77.79
GAT + ICL	91.02 ± 7.02	83.38 ± 1.7	75.12 ± 3.31	81.82 ± 1.20	72.60 ± 2.46	50.67 ± 3.60	78.98

Information-based Causal Learning (AAAI'24)

- Performance and Ablation study on the Synthetic Dataset and Real Dataset

Method	Spurious-Motif			MNIST-75SP	Graph-SST2	Molhiv	AVG
	b = 0.5	b = 0.7	b = 0.9				
Attention	39.42 ± 1.50	37.41 ± 0.86	33.46 ± 0.43	15.19 ± 2.62	81.57 ± 0.71	75.84 ± 1.33	53.87
Top-k Pool	41.21 ± 7.05	40.27 ± 7.12	33.60 ± 0.91	14.91 ± 3.25	79.78 ± 1.35	73.01 ± 1.65	53.94
SAG Pool	43.82 ± 6.32	40.45 ± 7.50	33.60 ± 1.18	14.31 ± 2.44	80.24 ± 1.72	73.26 ± 0.84	54.36
DIR	43.88 ± 4.27	41.87 ± 1.81	39.12 ± 3.51	19.47 ± 1.69	81.89 ± 0.73	68.04 ± 6.24	54.95
GCN	46.20 ± 2.34	38.12 ± 4.56	34.55 ± 1.23	11.35 ± 2.01	82.09 ± 3.45	95.79 ± 1.56	55.55
GCN + CAL	75.31 ± 11.35	69.38 ± 10.79	58.57 ± 5.94	15.76 ± 2.31	84.39 ± 0.28	96.59 ± 0.05	71.26
GCN + ICL	77.45 ± 12.45	75.09 ± 8.40	63.69 ± 8.73	17.04 ± 3.59	84.67 ± 0.37	96.89 ± 0.07	72.53
GIN	81.07 ± 3.12	69.30 ± 4.56	59.93 ± 2.34	11.80 ± 1.33	84.37 ± 2.56	96.84 ± 3.21	70.59
GIN + CAL	82.89 ± 8.53	86.86 ± 9.55	86.56 ± 8.91	18.74 ± 2.02	84.59 ± 0.33	97.19 ± 0.07	81.20
GIN + ICL	82.95 ± 8.53	89.00 ± 6.65	86.62 ± 5.40	19.07 ± 1.57	84.46 ± 0.46	97.19 ± 0.05	81.55
GAT	33.45 ± 2.12	33.60 ± 1.62	33.77 ± 3.45	9.80 ± 1.23	82.10 ± 4.56	97.01 ± 2.34	53.94
GAT + CAL	38.02 ± 6.87	39.42 ± 6.01	35.67 ± 4.35	20.64 ± 5.3	84.30 ± 0.4	97.24 ± 0.06	59.15
GAT + ICL	42.30 ± 10.29	42.01 ± 10.36	40.20 ± 7.78	21.29 ± 8.40	84.31 ± 0.4	97.27 ± 0.05	60.82





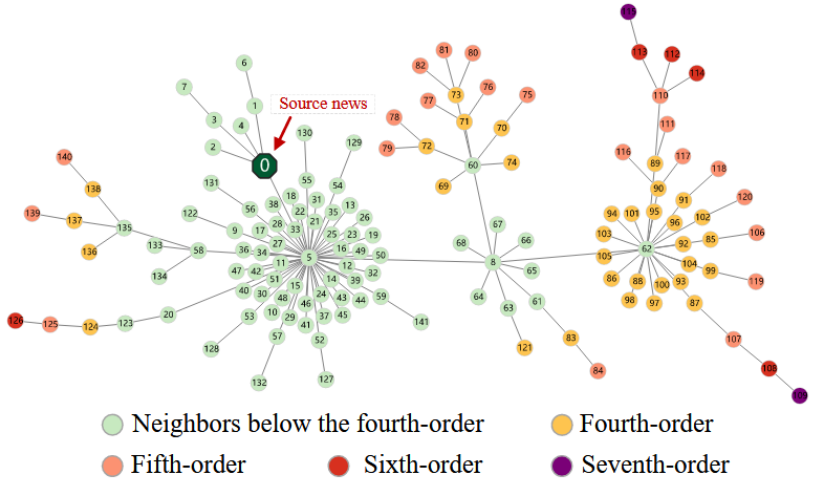
CONTENTS

1. What is Explainability?
2. Explainable GNNs
3. Identifying Informative Substructures
4. Conclusion

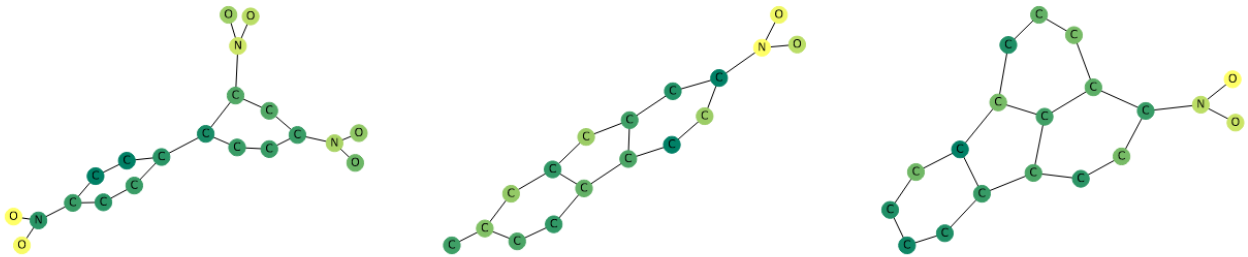
Recent trends

- Important nodes/edges/subgraphs

[1]



[2]



[3]

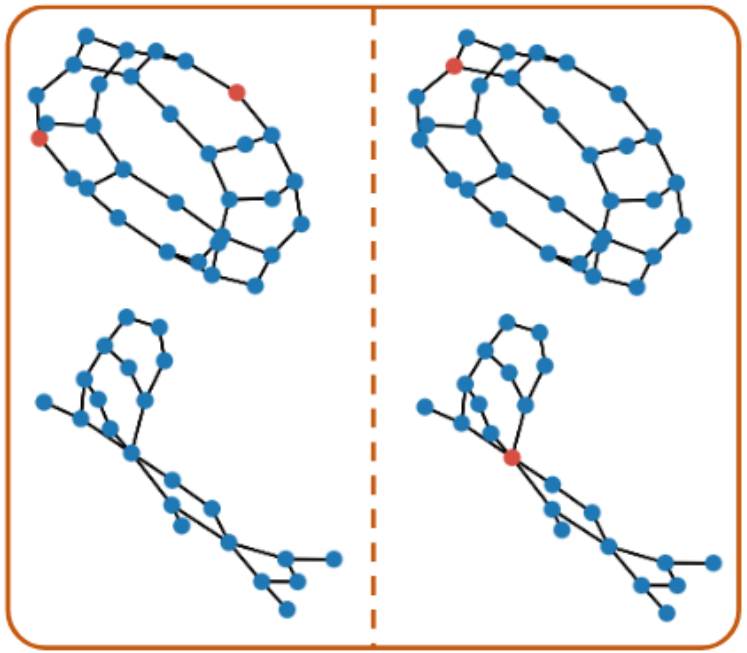
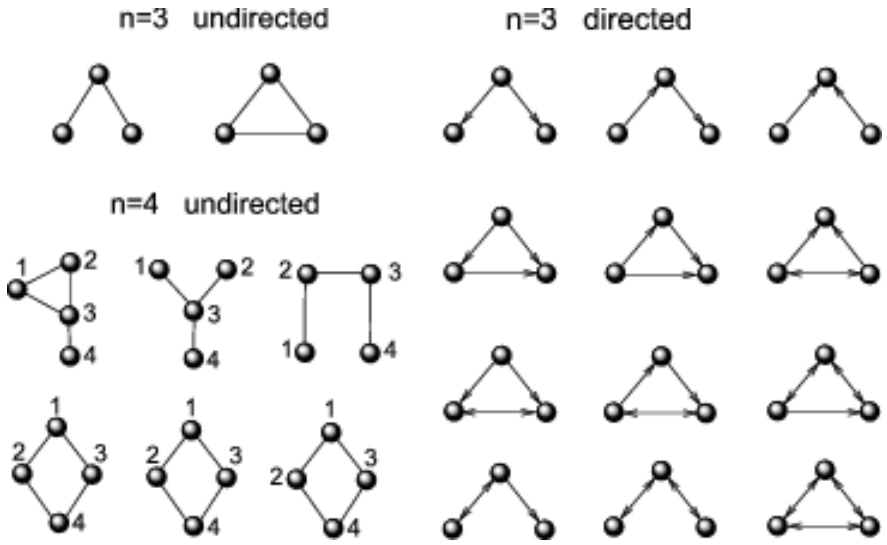


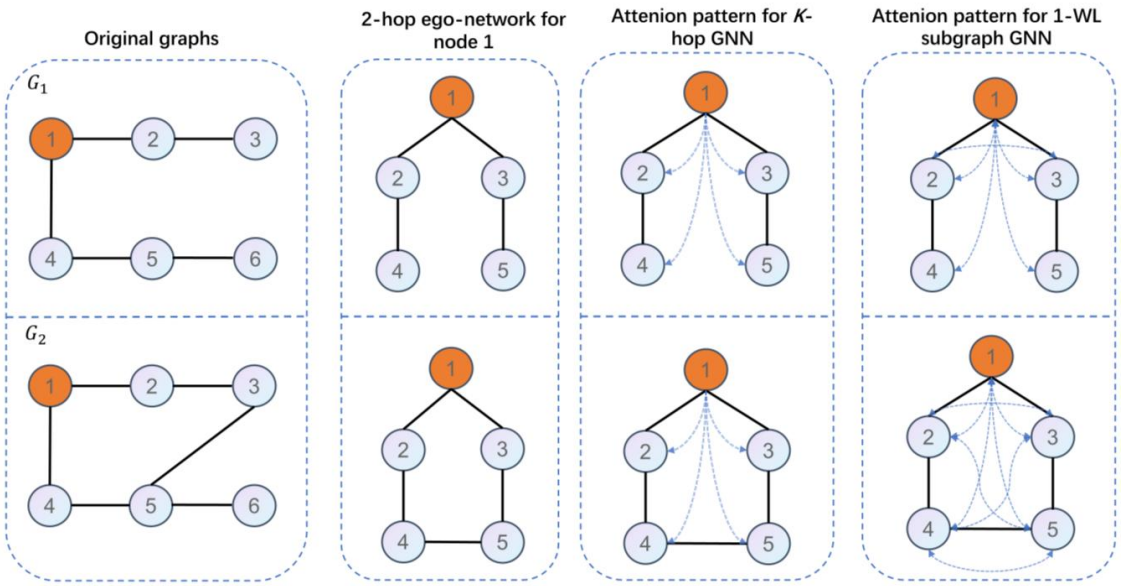
Figure 6: AgentNet node visitation heat map on test graphs from the MUTAG dataset. The brighter the color, the more often the given node has been visited. Agents prioritize some important substructures (NO_2) when they move around the graph.

[1] Propagation Structure-Aware Graph Transformer for Robust and Interpretable Fake News Detection, KDD'24
[2] Agent-based Graph Neural Networks, ICLR'23
[3] MAG-GNN: Reinforcement Learning Boosted Graph Neural Network, NIPS'23

Limitations

- Mismatch across representation units

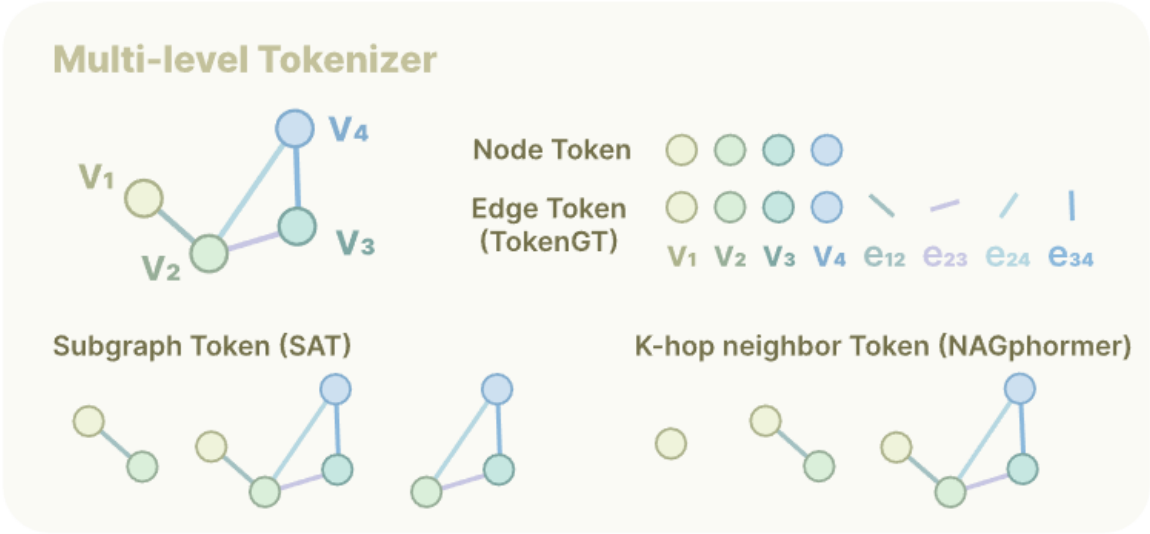




[1] https://www.mdpi.com/1099-4300/26/2/128
[2] Improving the Expressiveness of K-hop Message-Passing GNNs by Injecting Contextualized Substructure Information, KDD'23

Limitations

- Overlap / entanglement of substructures
  - ✓ Substructures가 겹치면 각 substructure의 기여/역할이 깔끔히 분리되지 않아 학습이 불안정해질 수 있음
- Heterogeneous / temporal graph에 적용하기 어려움
  - ✓ 해당 그래프에서는 "중요 구조"를 정의하는 것부터 애매함



[1] A Survey of Graph Transformers: Architectures, Theories and Applications, arXiv:2502.16533
[2] Evaluating the Structural Awareness of Large Language Models on Graphs: Can They Count Substructures?, KDD'24

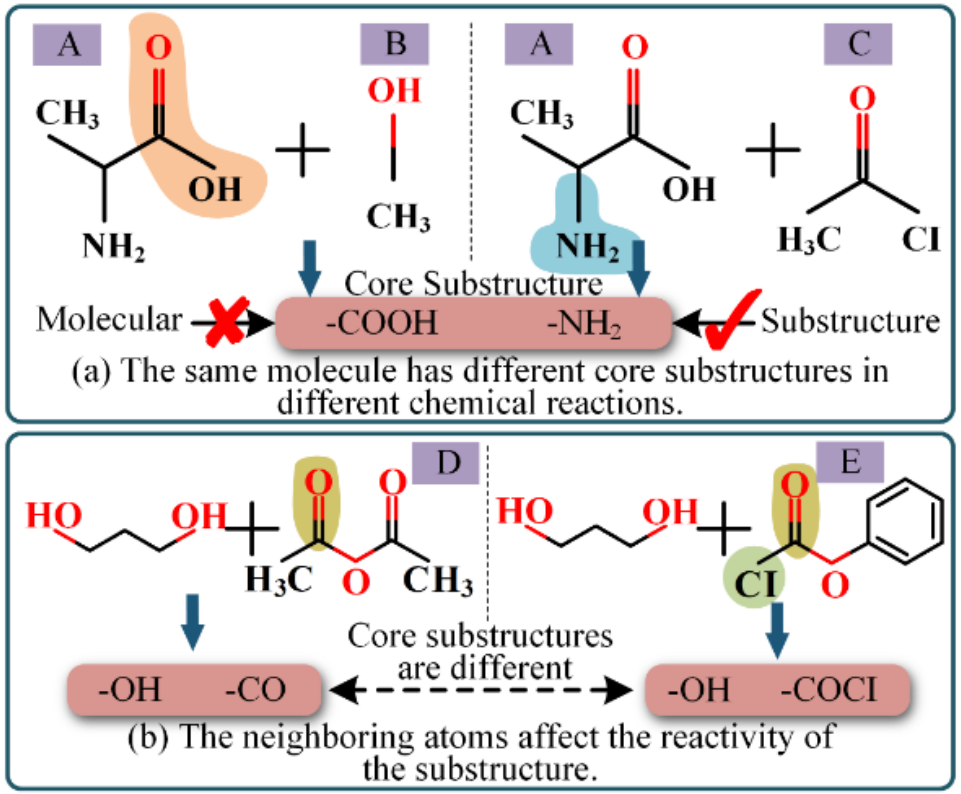
Limitations

- Substructure that is neither causal nor counfounding

[1] Subgraph Information Bottleneck with Causal Dependency for Stable Molecular Relational Learning, IJCAI'25
[2] Debiasing Graph Neural Networks via Learning Disentangled Causal Substructure, ICLR'22

Limitations

- Substructure that is neither causal nor counfounding



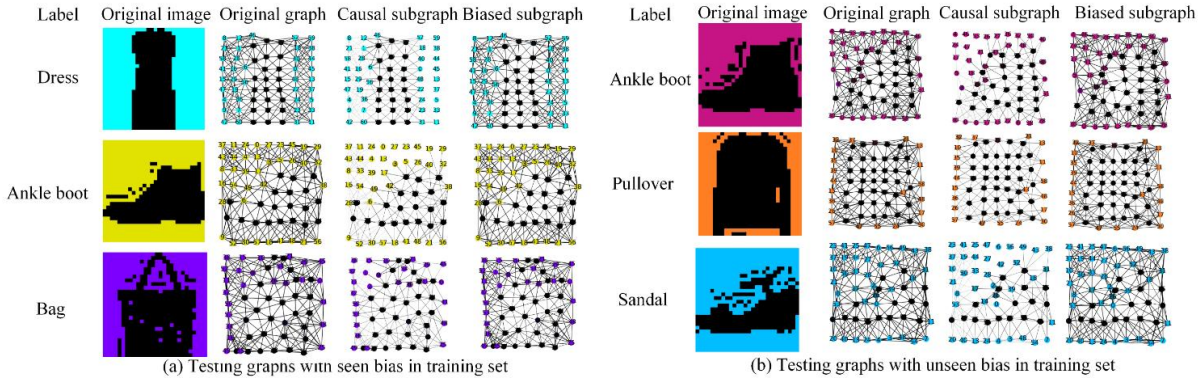


Figure 10: Visualization of subgraphs extracted by the mask generator from CFashion-75sp.

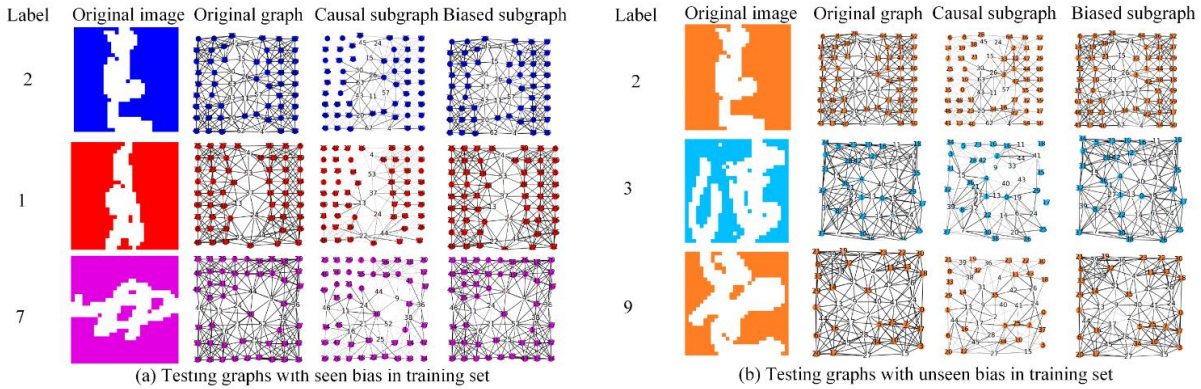


Figure 11: Visualization of subgraphs extracted by the mask generator from CKuzushiji-75sp.

[1] Subgraph Information Bottleneck with Causal Dependency for Stable Molecular Relational Learning, IJCAI'25
[2] Debiasing Graph Neural Networks via Learning Disentangled Causal Substructure, ICLR'22

Future works

- 여러 도메인에서의 causal subgraph 식별
  - ✓ not synthetic, but real-world dataset

- 점진적인 추론/샘플링을 통해 subgraph 선택 비용 절감

- Beyond homogeneous graphs
  - ✓ Temporal graphs
  - ✓ Heterogeneous graphs
  - ✓ Multimodal graphs

- More researches..
  - ✓ Graph Information Bottleneck
  - ✓ Subgraphormer (ICML'24) [1]
  - ✓ NeuralWalker (ICLR'25) [2]

[1] Subgraphormer: Unifying Subgraph GNNs and Graph Transformers via Graph Products, ICML'24
[2] Learning Long Range Dependencies on Graphs via Random Walks, ICLR'25

Thank you for listening ☺